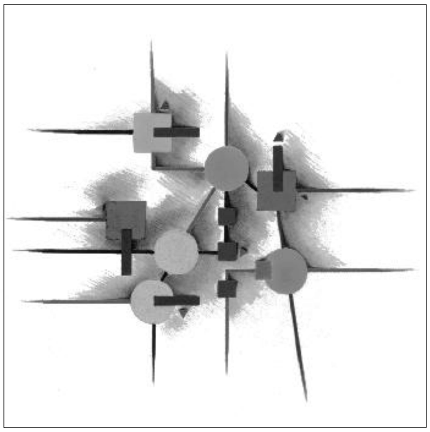
SAP Business Connector Security Best Practices



SAP SYSTEM

Release 4.8

SAP® AG • Dietmar-Hopp-Allee 16 • D-69190 Walldorf

Copyright

Icons

Icon	Meaning
⚠	Caution
🗨	Example
➡	Note
💡	Important

CONTENTS

Chapter 1: Introduction

Security is a critical part of all SAP BC server installations. This document provides a set of “best practices” to ensure that your SAP BC server is configured in the most secure way possible for your environment.

There is no single best configuration for all environments. Your optimal configuration will depend on your business needs, the tradeoffs between security and functionality, and the effort you want to spend implementing security capabilities. This document does not contain everything you need to know about security. Rather, it serves as a supplement to other SAP BC documents and your organization’s own security practices.

This document covers best practices for SAP BC server 4.8. Users of other SAP BC server releases should contact SAP Customer Care for the best practices information for those releases.

Principles Behind Best Practices

This document is based on a small set of widely accepted security principles:

- System access should be permitted only where explicitly granted; the default should be to deny access.

- Possible conflicts in permissions should be minimized.
- The system configuration should be as simple as possible.
- User groups should be set up for administrators, developers, and partners—to clearly separate system permissions—thus reducing the risk that an unauthorized user will gain access to a restricted capability.
- Users and software should be granted the fewest privileges necessary to accomplish their tasks.

Related Security Documentation

Along with this document, you should be familiar with the following chapters from the *SAP BC Administration Guide*:

- Chapter 2: "An Overview of the Server." The section entitled "Security Features" summarizes the security features that the SAP BC server provides.
- Chapter 3: "The Role of the Administrator." This chapter covers security information every administrator needs to understand.
- Chapter 6: "Configuring the Server."
    - "Configuring Ports" includes descriptions of how to configure the security parameters for each type of port.
    - "Allowing and Denying Inbound Connections to the Server" describes how to use SAP BC server capabilities to restrict which services an outsider can access (described more in this document).
    - "Specifying a Third-Party Proxy Server for Outbound Requests" describes how to configure your SAP BC server if you are using a proxy server, whether the proxy server is a SAP BC server or some other type of proxy.
- Chapter 7: "Managing Users and Groups." This chapter describes how to set up users and groups, and their relationship to Access Control Lists (ACLs).
- Chapter 8: "Managing Server Security." This key chapter describes how to use key security mechanisms such as

ACLs, how to set up SSL, and how to limit access to services by port.

- Chapter 9: "Using an External Directory (LDAP or NIS)." This chapter describes how to configure authentication to use an LDAP or NIS server.
- Appendix A: "SAP BC server Deployment Checklist." This appendix includes a checklist for fielding a SAP BC server. In particular, stage 7 includes useful security hints.
- Appendix B: "Server Configuration Parameters." The section entitled "watt.security" identifies configurable security parameters stored by the SAP BC server.
- Appendix C: "Server Log Files." This appendix defines the structure of the system log files, including the audit file.

Security Background Information

You may also find the following books useful in understanding security concepts:

- *Practical UNIX & Internet Security, 2nd edition*, Simson Garfinkel and Gene Spafford, O'Reilly & Associates, 1996.
- *Firewalls and Internet Security, 2nd edition*, William Cheswick and Steven Bellovin, Addison-Wesley, 2001.
- *Internet Cryptography*, Richard Smith, Addison-Wesley, 1997.
- *Secure Electronic Commerce, 2nd Edition*, Warwick Ford and Michael Baum, Prentice Hall, 2000.
- *Network Security Essentials: Applications and Standards*, William Stallings, Prentice Hall, 2000.
- *SSL and TLS : Designing and Building Secure Systems*, Eric Rescorla, Addison-Wesley, 2000.

Disclaimer

This document is provided for informational purposes only, and is not intended to, nor shall it create any warranty, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

This document does not constitute "documentation" for purposes of any Software License Agreements between SAP and its customers and partners.

SAP shall not be liable for any damages, including but not limited to indirect, direct, special, incidental, or consequential damages arising out of this document or the information contained herein, even if SAP has been advised of the possibility of such damages, and notwithstanding any failure of essential purpose of any limited remedy of any kind.

Mention of a product in this document does not imply endorsement by SAP. SAP BC Developer and SAP BC server are trademarks of SAP. SAP and the SAP logo are registered trademarks of SAP. All other marks are the property of their respective owners.

Chapter 2: Before You Begin

This section identifies several issues to consider before you begin implementing the best practices described in this document.

Determining Your Security Posture

The first thing you need to do is to determine your security posture. This posture applies not only to your SAP BC server, but also to all computer systems in your environment. The following are general characterizations; select the one that is closest to your organization's security posture. Determining your security posture will guide you through the rest of this document—it will help you decide which of the security best practices to follow.

🔒🔒🔒🔒 **Very High:**

- Your SAP BC server is connected to the Internet, and your organization has strong policies about security configurations.

- You are concerned about attacks against your SAP BC server from both inside and outside your organization.

- As a policy, you want to minimize the risks of accidentally malfunctioning software.

- By default, you want to disallow all access not specifically allowed.

- You plan to use HTTPS for nearly all of your SAP BC server communications, and you will use 1024- or 2048-bit public keys with 128-bit private keys for maximum SSL protection.

- You plan to avoid using FTP, HTTP, and mail for SAP BC server communications.

- Your network configuration includes filtering routers, firewalls, and a Demilitarized Zone (DMZ), with one or more proxy servers in front of any SAP BC server or application servers.

- You recognize significant risks to your services, and are willing to put in the amount of effort required to maximize the security of all of your systems.

🔒🔒🔒 **High:**

- Your SAP BC server is connected to the Internet, and your organization has policies about security configurations.
- You are concerned about attacks against your SAP BC Server, mostly from outside your organization, and to a lesser extent, from inside your organization.
- By default, you want to disallow all access not specifically allowed.
- Your network configuration includes filtering routers or firewalls and a Demilitarized Zone (DMZ), with one or more proxy servers in front of any SAP BC server or application servers.
- You plan to use HTTPS for nearly all of your SAP BC server communications, and you will use 1024- or 2048-bit public keys with 128-bit private keys for maximum SSL protection.
- You plan to avoid using FTP, HTTP, and mail for SAP BC server communications.
- You perceive significant risks to your services, and are willing to put in some extra effort to ensure the security of all of your systems.

🔒🔒 **Medium:**

- Your SAP BC server is connected to the Internet, and your organization might have some policies about security configurations.
- You are concerned about attacks against your SAP BC server from outside your organization, but not from inside your organization.
- You might want to allow all access not specifically denied.
- Your network configuration includes filtering routers or firewalls and may have a Demilitarized Zone (DMZ).

- You plan to use HTTPS for some of your SAP BC server communications, and you will use 1024-bit public keys with 128-bit private keys for maximum SSL protection.

- If you use HTTP, FTP, or mail for some SAP BC server communications, it is restricted to non-sensitive data or it is protected using a Virtual Private Network (VPN).

- You perceive moderate risks to your services, and are willing to put in a moderate amount of effort to ensure the security of your SAP BC server.

🔒 **Low:**

- Your SAP BC server is in a test environment, or if it is connected to the Internet, contains no critical information.

- Your organization might some have policies about security configurations.

- You are not concerned about attacks against your SAP BC server from inside your organization.

- By default, you want to allow all access not specifically denied.

- You will use HTTPS, HTTP, FTP, or mail for your SAP BC server communications, and are relatively unconcerned with protecting data in transit.

- Your network configuration includes a firewall, but probably does not have a DMZ, or your system might not be connected to the Internet.

- You perceive little risk to this system, and want to minimize the extra effort required to ensure the security of your SAP BC server.

What You Need Before You Start

Before you start reading the security best practices in this document, you need to obtain the following information:

- A list of all application-specific services in your SAP BC server, and whether they are directly callable or for internal use within the server. For directly callable services, determine who should be able to access them (for example, administrators or trading partners).

Chapter 3: Overview of Best Practices

This section summarizes the best practices and indicates which best practices you should follow, depending on your security posture, as described in the previous section. The first column of the table includes one through four padlock symbols, to be interpreted as follows:

🔒 🔒 🔒 🔒	Implement if your security posture is very high.
🔒 🔒 🔒	Implement if your security posture is high or very high.
🔒 🔒	Implement if your security posture is medium, high, or very high.
🔒	Implement if your security posture is low, medium, high, or very high.

These are general suggestions—your organization may differ in the importance given to particular measures. As you review this list, you may want to check the boxes (□) you plan to implement. When you finish updating the server to meet the best practices, check this list again to make sure you have performed all of the checked steps.

Security Posture	Security Measure	See Page
🔒	1. □ Apply these best practices to the SAP BC server in your test environment before applying to a production server.	43
🔒 🔒	2. □ Make a backup of the entire SAPBC file system before applying these best practices. You may need this backup in case of later configuration errors.	N/A
🔒	3. □ Using Basic Infrastructure Protections	13
🔒 🔒	3.1. □ Examine the placement of your SAP BC server(s) within your Intranet and/or DMZ.	13
🔒 🔒	3.2. □ Check for security-related patches or new versions of the OS and Java VM.	13
🔒 🔒	3.3. □ Use firewall and router rules to allow access to only those ports used by the SAP BC server.	13
🔒	3.4. □ Do not install the SAP BC server on the same computer as the firewall.	13
🔒	4. □ Selecting a Secure Transmission Protocol	14
🔒 🔒	4.1. □ If possible, use HTTPS, or consider a VPN with any other protocol.	14
🔒 🔒 🔒	4.2. □ If internal access is a concern, use the firewall to constrain it.	15
🔒 🔒 🔒	4.3. □ (For UNIX) If you want to use default port numbers, use port remapping to avoid running as root, or consider using a higher numbered port.	15
🔒 🔒 🔒	4.4. □ If possible, run the SAP BC server as a user ID with permissions to access its own files, but no other files on the system.	15
🔒	5. □ Authentication Mechanisms	16
🔒	5.1. □ Create new users to replace the built-in Administrator, Developer, and Replicator accounts and disable the built-in accounts.	16

Security Posture	Security Measure	See Page
🔒	5.2. □ New users that are members of the Developers group should set secure passwords themselves.	17
🔒	5.3. □ For all other users, the Administrator should select secure passwords.	17
🔒 🔒	5.4. □ Create individual accounts for all users, including Administrators and Developers (rather than sharing the predefined accounts among multiple users).	17
🔒 🔒 🔒	5.5. □ If using locally stored passwords (that is, not LDAP), use OS permissions to protect the password file `<`*`sapbc`*`>\server\config\users.cnf`. Restrict access to this file to only the user ID that runs the SAP BC server.	18
🔒 🔒	5.6. □ Although the Administrator can set minimum password characteristics, do not set them any lower than the system defaults.	18
🔒 🔒	5.7. □ When mapping digital certificates to users, assign to the least privileged IS user name possible for that user’s responsibilities.	18
🔒 🔒 🔒	5.8. □ Use SSL and require client certificates. Use OS permissions to protect private key files.	18
🔒 🔒 🔒	5.9. □ Verify that `watt.security.ssl.ignoreExpiredChains` is set to “false” (default value). If this setting is not in the configuration file, add it.	19
🔒 🔒 🔒	6. □ Controlling by IP Address or Domain	21
🔒 🔒 🔒	6.1. □ Configure both the firewall and the SAP BC server settings to restrict access to your trading partners.	22
🔒 🔒 🔒	6.2. □ Use the Allowed Hosts list to allow access only to those hosts of your partners that should be connecting to your SAP BC server.	22
🔒 🔒 🔒	6.3. □ If possible, use IP addresses (harder to spoof) and/or configure your firewall to detect and reject domain name spoofing attempts.	23
🔒 🔒 🔒	6.4. □ When using one SAP BC server in your DMZ and one in your intranet, configure the intranet server to only allow access from the server in the DMZ.	22
🔒 🔒	7. □ Verify the factory service/folder ACL settings. (Refer to Appendix A in this document and the corresponding *Best Practices* documents for other SAP products.)	28
🔒 🔒	8. □ Review the suggested changes to the default service/folder ACL settings. Check for dependencies and don’t make the changes if your application may be affected adversely. (Refer to Appendix C in this document and the corresponding *Best Practices* documents for other SAP products.)	28
🔒 🔒	9. □ Review all of your custom services/folders and apply the appropriate ACL: Administrators, Developers, Default, Anonymous, or custom.	29
🔒 🔒 🔒	10. □ Set up an Administrator port.	30
🔒 🔒 🔒	11. □ Set up a Developers port.	32
🔒 🔒 🔒	11.1. Disable the Developer user.	32
🔒 🔒 🔒	12. □ Set up a Replicators port.	37
🔒 🔒 🔒	12.1. □ Disable the Replicator user. Publishers and subscribers can have their own user names.	36
🔒 🔒 🔒	12.2. □ Use SSL for package distribution.	37
🔒	13. □ Protect Internal Services from External Access (Port Controls), for each port if possible.	38
🔒 🔒	13.1. □ Apply appropriate firewall and IP/Domain protection.	21
🔒	13.2. □ Identify which services/folders should be available for external access.	38

Security Posture	Security Measure	See Page
🔒	13.3. □ Add the list of external services needed by adapters and layered products to the port's Allow list. (Refer to Appendix F in the *Best Practices* documents for the adapters and other SAP products.) Enter service/folder names in the form `folder1.folder2.folder3:service.`	39
🔒	13.4. □ Add your list of application-specific services that are accessible from each port.	39
	13.5. □ Update the IP addresses for the port, specifying allowed only hosts if possible.	22
🔒 🔒	14. □ Protect DSPs and other files in the pub directories.	40
🔒 🔒	14.1. □ Check the factory settings for built-in pub files. (Refer to Appendix D in this document and corresponding *Best Practices* documents for other SAP products.)	41
🔒 🔒	14.2. □ Ensure that the \pub directories of all packages, especially those you created yourself, are protected by a `.access` file. Ensure that all subdirectories of \pub are also protected by a `.access` file.	42
🔒 🔒 🔒	14.3. □ Either set the `watt.server.displayDirectories` parameter to "false" or create an `index.html` file for every \pub directory.	42
🔒 🔒	14.4. □ Reload the package for `.access` file changes to take effect. Restart the server if you made changes to `.access` files in the WmRoot package.	42
🔒 🔒	15. □ Transition from Testing to Production	43
🔒 🔒	15.1. □ For a production system, disable Developer access by removing all groups from the Developers ACL or removing as many users as possible from the Developers group. In addition, disable Internal access by removing all groups from the Internal ACL or removing as many users as possible from the Internal group.	43
🔒 🔒 🔒	15.2. □ Delete or disable the WmSamples package.	44
🔒 🔒 🔒	16. □ Auditing and Forensics	44
🔒 🔒 🔒	16.1. □ Archive audit logs by backing them up and storing them off line.	44
🔒 🔒 🔒	16.2. □ Before going into production, calculate cryptographic checksums of all files that should not change. Check for modifications periodically.	44
🔒 🔒	16.3. □ Before going into production, make a complete backup of the SAP `BC` directory and store off line. Make a new backup whenever the configuration changes.	44
🔒 🔒 🔒	16.4. □ Create and review audit logs regularly.	45
🔒	17. □ Test the new configuration, especially using different user IDs. Check for access to the proper services on each port and for unauthenticated access.	43
🔒	18. □ Propagate changes to other SAP BC servers in your environment.	45

Chapter 4: Using Basic Infrastructure Protections

The SAP BC server is part of your total Information Technology infrastructure. As such, security for the server is affected by your overall system architecture.

A SAP BC server can be placed in your Demilitarized Zone (DMZ), in your Intranet, or both. An alternative to using a SAP BC server in the DMZ is to use a *proxy server*, such as IPlanet's IPlanet Proxy Server, in the DMZ. Another option is to have a SAP BC server in the DMZ connecting directly to your intranet systems; this configuration is less common, and is not recommended for security-conscious customers.

Best practice: Place the SAP BC server in your Intranet, and use a SAP BC server or a proxy server in the DMZ to perform initial filtering of requests. Never place a SAP BC server outside your outer firewall.

A SAP BC server is only as secure as the operating system it runs on. As part of securing your server, verify that the underlying operating system (such as Windows, Solaris, or HP-UX) is configured securely, and that you have installed all security patches. You should also install all patches for your Java Virtual Machine (JVM), because flaws in the JVM can lead to security problems for the server.

Best practice: Check regularly with your operating system and JVM vendors for any patches, and install them promptly. Turn off all unneeded network services, such as TELNET servers. Use security assessment tools to determine whether there are any unpatched vulnerabilities. Commercial tools in this category include Internet Security Systems' Internet Scanner (www.iss.net) and Network Associates' CyberCop Scanner (www.pgp.com). Freeware tools in this category include Nessus (www.nessus.org). There are also useful tools for verifying the correct configuration of your operating system.

Best practice: Regardless of where you place your SAP BC server, firewalls and filtering routers are a key part of protecting the server. Be sure that you allow access to only those ports that the server is actually configured to use.

Best practice: Do not install the SAP BC server on the same machine that serves as a firewall—neither is intended to be used in that way. This configuration increases the risks associated with any malfunction of the server and reduces the performance of the combined system.

When installing your SAP BC server, consider the TCP/IP ports it will use and how that relates to the rest of your infrastructure. The following section discusses the tradeoffs in selecting and setting up ports.

Chapter 5: Selecting a Secure Transmission Protocol

SAP BC server supports several different transport protocols for communication between SAP BC servers or between one SAP BC server and other types of clients or servers: HTTP, HTTPS, FTP, and E-mail (SMTP for outbound; POP and IMAP for inbound).

Best practice: Customers concerned with security should use HTTPS, because it provides SSL protection of the traffic. If HTTPS is not an option, you can use a Virtual Private Network (VPN) with any of the other protocols to provide protection in transit. To minimize exposure, configure the VPN to allow only the required protocol (for example, FTP) between the ends of the communication path, rather than opening the two networks to each other. Opening the network completely makes each end vulnerable to all attacks on the other.

VPN encryption is comparable to HTTPS encryption. However, HTTPS has the advantages of protecting the data all the way to the SAP BC server and providing application-level authentication through SSL client certificates. VPN encryption, on the other hand, has some disadvantages: the data is decrypted at the VPN device, then passed unprotected to the SAP BC server; and there is no application-level authentication because the VPN does not send any authentication information to the application.

➡ **Note:** You may use nCipher's nFast, nForce, and nShield products to minimize the performance impact of using SSL. See www.nCipher.com for information about these products. For information about configuration, see "Securing Communications with the Server" in Chapter 8 of the *SAP BC server Administrator's Guide.*

When establishing a connection using HTTPS, a SAP BC server will use the strongest encryption suite it can, preferably 128-bit encryption. If the receiving server is only capable of "weak" encryption (for example, it uses a 40-bit session key), the server will accept a weak connection.

Best practice: You can purchase digital certificates for use with SSL with different levels of encryption. Be sure to purchase 1024-bit (or greater) certificates with support for 128-bit sessions. While these certificates are more expensive than other types, they provide maximum protection for your data.

The SAP BC server can use the standard port numbers for each of its protocols: 21 for FTP, 25 for e-mail, 80 for HTTP, and 443 for HTTPS. By default, it uses port 5555 for HTTP and 8021 for FTP. There are no ports preconfigured for any other services.

Best practice: Use your organization's firewall to reinforce the selection of a transmission protocol. Configure the firewall so that connections from the outside to the SAP BC server can access only those ports you select, such as 443 for HTTPS.

Best practice: If you are concerned about connections from the inside (that is, from within your organization's internal network) to the SAP BC server, use your firewall to control access from the *inside* to your SAP BC server as well.

Running your SAP BC server on a UNIX-Based System

If your SAP BC server is running on a UNIX-based system (such as Solaris, HP-UX, Linux, or AIX) *and* you want to use port numbers below 1024 (e.g. the standard ports mentioned above), you run into a problem: on UNIX systems only a *root* user is allowed to open ports below 1024. However running the SAP BC (and thus the JVM) as *root* is not recommended, as a successful attack on the BC or the JVM could give the attacker *root* access to the system! For this reason it is not recommended to start the BC under *root* (e.g. by (1) making the JVM executable *setuid-root*, (2) starting the JVM while logged in as root, or (3) starting the JVM as root from a system startup file). Instead use one of the following best practices:

Best practice: You can minimize the overall risk by not running the JVM as root or using port remapping, and instead exposing higher numbered ports (1024 or above) to the outside world.

Best practice: If exposing higher number port numbers is not acceptable (that is, you need to use default port numbers), and your host operating system supports port remapping (e.g. the external users can access the default port, and the operating system forwards the request to your server using a higher numbered port), use this technique instead of running the JVM as root. Running the JVM as root is riskier than using port remapping, since a flaw in the JVM could lead to remote root access.

Best practice: If exposing higher port numbers is not acceptable, and your host operating system does not support port remapping, but your firewall provides this feature, use this technique instead of running the JVM as root.

Best practice: Some UNIX systems allow you to specify additional users beside the root user, which shall be authorized to open a port below 1024. Consult your UNIX manual, and if your host system provides this feature, add the user, under which the SAP BC will be started, to the list of these privileged users.

The following best practice should be implemented in any case:

Best practice: If possible, run your JVM as a user ID with permissions to access the SAP BC server's files, but no other files in the system. By doing so, you reduce the risk of a flaw that could allow a remote user to run an external program. For example, rather than invoking the `server.sh` script directly from the system startup script, use the command `su bcuser` `server.sh`, and give the user `bcuser` access only to the JVM and SAP BC server files.

Running your SAP BC server on a Windows System

If your SAP BC server is running on a Windows XP/Vista/7 or Windows Server 2003/2008 system, you can use the default ports without any special configurations.

Best practice: No special port permissions are required to run the SAP BC server. However, it is best to run the server as an identity other than Administrator, and as a user ID with permissions to access the server's files, but no other files in the system. By doing so, you reduce the risk of a flaw that could allow a remote user to run an external program. For example, it is not recommended that you run the `server.bat` script directly as part of the Windows bootup process. Instead, you could create an unprivileged user `bcuser` that has access only to the JVM and SAP BC server files, then use an automatic login as `bcuser`, and have that user's login script start the server.

Choosing the Best Ports

There is no inherent security advantage to using either standard or non-standard ports. The main caveat is to avoid running the JVM as root. Attackers routinely use port scanners to look for services, so "hiding" on obscure ports is of little value. Instead, when you are selecting ports, you should focus on operational needs and ease of firewall configuration.

Chapter 6: Authentication Mechanisms

Clients or servers can prove their identity to a SAP BC server using either a user name/password combination or using SSL client certificates with a corresponding private key.

User Name/Password Authentication

The user name/password mechanism is covered in detail in Chapter 7 of the *SAP BC server Administrator's Guide*. Additionally, Chapter 9 of the *SAP BC server Administrator's Guide* describes how to use an LDAP or NIS server to store user names and passwords, rather than storing them locally on the server.

Best practice: The SAP BC server comes with several built-in accounts: Administrator, Developer, and Replicator. Be sure to change the passwords for all three accounts as soon as you complete the installation. These are very powerful accounts, so be sure to set secure passwords. In addition, depending on the adapters or layered product you use, there may be additional built-in accounts. You should also change these passwords immediately after installation.

Note: While the Administrator and Developer accounts (and the associated Administrators and Developers groups) have different default capabilities, they have essentially equivalent privileges. A member of the Administrators group can gain all privileges associated with the Developers account, and vice versa. You should not give users the passwords to either of these built-in accounts or assign them to either of these groups unless you can trust them to safeguard your organization's information. You should never assign trading partners to either of these groups.

Best practice: Instead of using the built-in accounts, it is better to create an individual account for each administrator and developer, and then disable the built-in account. Advantages of individual accounts include:

- Ease of changing passwords: There is no need for sophisticated mechanisms to distribute modified passwords.
- Ease of revoking access: An individual user's access can be revoked by disabling or deleting the account, rather than having to notify all users of the shared account what the new password is.
- Individual accountability: Audit logs will show what actions were taken by each user, rather than by any user who has the shared password.
- Less predictable account names: If you disable the built-in accounts after creating individual accounts, an attacker will be unable to break in using any of the built-in accounts, and will have to determine both the password and user name.

Best practice: When you create local a user account (that is, an account stored on the SAP BC server and not in an LDAP or NIS database), be sure to specify a password at the time you create the account. Otherwise, there could be a time interval in which an attacker could use the account to gain access to your server. Minimize this interval by setting the password before performing any other functions.

Caution: For *SAP BC* 3.1 and earlier, for replication to occur, the Replicator account for all SAP BC servers participating in the exchange must have the *identical* password. For *SAP BC* 3.5 and later, each server can have its own Replicator account password.

Best practice: If using user names and passwords, allow only members of the Developers group to change their own passwords. The administrator must select secure passwords for all other users, and developers must select secure passwords for themselves.

Note: If you choose to store your usernames and passwords in an LDAP server, be aware that communications with the LDAP server are not encrypted at the protocol level, so be sure to select the “crypt” hashed passwords option when configuring your LDAP server (so that passwords will not flow across the network in the clear).

Note: If your SAP BC server is in the DMZ and you are using LDAP or NIS to store usernames and passwords, you will also need access to the LDAP/NIS server in the DMZ. If that server is inside your inner firewall, you will need to open a port to allow access through the firewall to the LDAPR/NIS server, which introduces some security risk.

Best practice: The administrator can set minimum password characteristics for passwords set by developers. SAP suggests that you use the default password requirements as a *minimum*. The defaults are a minimum password length of eight characters, with at least two uppercase and two lowercase letters, one numeric, and one “special” character.

Private Key/Certificate Authentication

Chapter 8 of the *SAP BC server Administrator’s Guide* describes how the set up your server to use of SSL client certificates for authentication.

Best practice: When setting up the mapping of SSL certificates to users, be sure to assign the user identity to the least privileged user name commensurate with the user’s responsibilities. Just because users have authenticated with a certificate does not mean they should have unlimited access to the SAP BC server.

Best practice: The most secure method of authentication is using SSL client certificates with the corresponding private key. Unlike passwords, the secret portion of the authentication (your private key) is never transmitted to the host to whom you are authenticating yourself. However, the private key is stored in the file system, so you should operating system file permissions to protect it. The key file should be readable *only* by the identity used to run the SAP BC server, but not any other programs on the system.

All SSL client certificates are good for a fixed period of time. The most secure setting (which is the default) is that the server will reject an expired certificate (including any signing certificate in a certificate chain). You can control this behavior by setting the `watt.security.ssl.ignoreExpiredChains` parameter in the server configuration file to “false” (the default, secure setting) or “true” (a less secure setting).

Best practice: By default, the `server.cnf` file does not include an explicit setting for the `watt.security.ssl.ignoreExpiredChains` parameter. Add it to the configuration file with the value "false".

Chapter 7: Access Control Mechanisms

Your SAP BC server provides access to several kinds of resources:

- *Services*, which are executable code that take parameters from users, perform some actions, and (typically) generate a response. *Folders* are logical groupings of services.

- *Dynamic Server Pages (DSPs)*, which are dynamically created web pages that a browser returns and displays to a user. DSPs configure presentation logic with service invocations.

- Files within the *pub* directory tree of the SAP BC server's packages.

The methods for controlling access to these types of resources are closely related, but not identical, as you can specify separate controls for services versus DSPs and files.

Note: The SAP BC server does not support Java servlets, CGI scripts, NSAPI plugins, or ISAPI plugins; therefore, security issues associated with those technologies are irrelevant.

There are several levels of protection for your SAP BC server.

1. **Firewalls:** Your firewalls and filtering routers prevent the attacker from outside your organization from accessing your SAP BC server. Depending on your network configuration, you may also use firewalls and filtering routers to prevent attacks by insiders.

2. **SSL:** SSL prevents an attacker from reading traffic between your SAP BC server and other servers.

3. **SAP BC server allow/deny by IP/domain:** You can configure the SAP BC server to disallow access by IP addresses or domains outside a specified list.

4. **Port controls allow/deny:** You can configure the SAP BC server to prevent both authenticated and unauthenticated users from accessing internal services that are used by the externally visible services. This mechanism is known as *port controls*. For example, even if a purchase order submission accesses a database, external users should not be able to access the database directly.

5. **Access Control List (ACL) configurations:** Security features within the SAP BC server can further limit the services that groups of users can access. For example, you may have some trading partners who can query a catalog, while other partners can both query the catalog and submit purchase orders. You may place similar controls on which groups of users can access particular DSPs.

6. **Application restrictions:** The SAP BC server application itself can enforce restrictions based on application-specific criteria. For example, your flow services can restrict the dollar value of a purchase order based on criteria in your database. These may be business rules or part of the security policy. You must implement such limitations in your SAP BC server application; they are not part of the core capabilities of the SAP BC server.

You can implement items 4 and 5 by dividing the set of services that the SAP BC server provides into several groups:

a. **Administration:** Services that are needed only by administrators of the SAP BC server.

b. **Anonymous:** Services that must be available to unauthenticated users so that they can connect, disconnect, and use guaranteed delivery services.

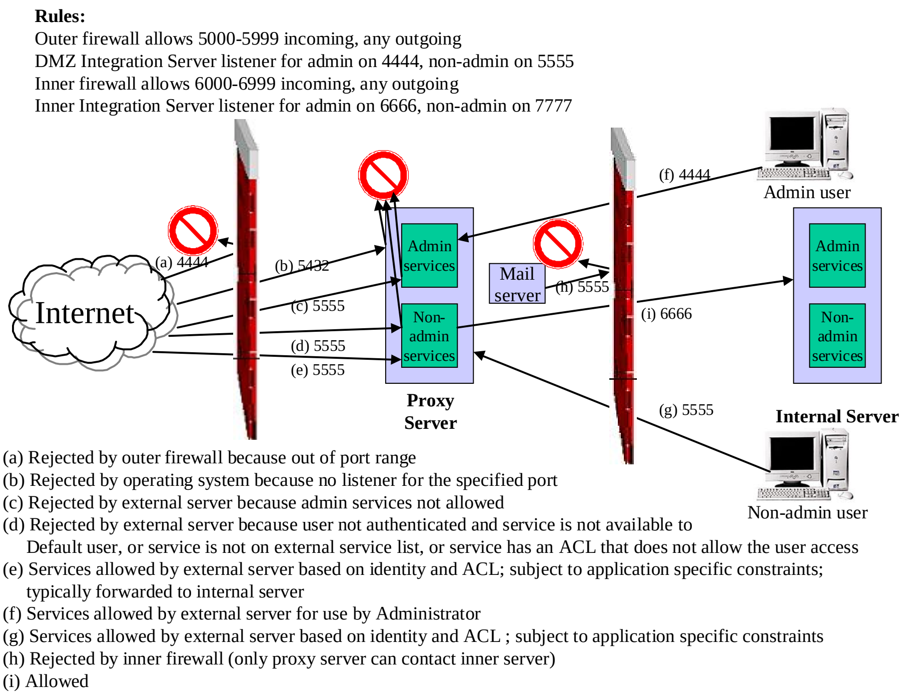
c. **Entry point:** Services that need to be available to external users. Your SAP BC server might offer features such as submitting purchase orders, entering invoices, and retrieving catalog entries.

d. **Internal:** Services that need to be available inside the server, but should not be available to external users. These include the lower level functions used to implement the external services. For example, services to access databases or adapters to internal systems should be in this group.

e. **Developer:** Services that are needed to develop applications for the SAP BC server, but should not be available to non-developer users and generally not accessible on a deployed server.

f. **Replicator:** Services that are needed to replicate packages from one SAP BC server to another, but should not be available to users.

Figure 1 shows the network infrastructure of the SAP BC server and how it provides these protections.



Rules:
Outer firewall allows 5000-5999 incoming, any outgoing
DMZ Integration Server listener for admin on 4444, non-admin on 5555
Inner firewall allows 6000-6999 incoming, any outgoing
Inner Integration Server listener for admin on 6666, non-admin on 7777

(a) Rejected by outer firewall because out of port range
(b) Rejected by operating system because no listener for the specified port
(c) Rejected by external server because admin services not allowed
(d) Rejected by external server because user not authenticated and service is not available to Default user, or service is not on external service list, or service has an ACL that does not allow the user access
(e) Services allowed by external server based on identity and ACL; subject to application specific constraints; typically forwarded to internal server
(f) Services allowed by external server for use by Administrator
(g) Services allowed by external server based on identity and ACL ; subject to application specific constraints
(h) Rejected by inner firewall (only proxy server can contact inner server)
(i) Allowed

Figure 1. Protections in a SAP BC server Environment

The following sections describe how to implement each of these facilities, with the exception of SSL and firewall services. In addition, it describes how to set up `.access` files to control access to Dynamic Server Pages served by the SAP BC server.

Controlling by IP Address or Domain

This portion of the instructions is optional, but highly recommended.

The first step in controlling access to your SAP BC server is determining which IP addresses and/or Internet domains can access the server. For example, if you know that you will be communicating with company1.com and company2.com, but no others, you should start by disallowing connections from all except those domains. Similarly, if you know the IP addresses (either specific addresses or ranges) that you communicate with, you could limit connections to those addresses. Note also that you can configure this access by individual port. For example, the port for administrators can be restricted to internal IP addresses, whereas the application port can be opened to trading partner IP addresses.

You can control access by IP address or domain by using your firewall, filtering router, and/or the server itself. Consult your firewall or filtering router documentation for information about how to use those capabilities. To control access using the SAP BC server, see “Allowing and Denying Inbound Connections to the Server” in Chapter 6 of the *SAP BC server Administrator’s Guide*.

The SAP BC server allows you to specify either a list of hosts (or IP addresses) to be allowed access or a set to be denied access. Setting a port type to Deny By Default allows an administrator to specify an Allowed Hosts list. Only the hosts in this list are allowed access via that port. Conversely, if a port is configured as Allow By Default, the administrator may specify a Denied Hosts list. In this mode all addresses except those listed are allowed.

Best practice: It is safer to specify an Allowed Hosts list than a Denied Hosts list, under the security maxim of "that which is not (explicitly) allowed is denied." However, if it is not feasible to have an Allowed Hosts list, a Denied Hosts list is better than no list at all. You might want to list domains in your Denied Hosts list that are unlikely to be legitimate trading partners, such as domains that are used for hosting home users, countries with which you do not ordinarily do business, competitors, etc. While this will not preclude an attack from these domains, it is a prudent measure and reduces the threat.

Best practice: Use the Allowed Hosts list in the SAP BC server (and the corresponding firewall or filtering router rules) to allow access *only* by those hosts of your partners that should be connecting to the server. For example, rather than allowing "*.mypartner.com", allow ??"is.mypartner.com"??. If you prefer to use IP addresses rather than host names, it is preferable to allow "127.10.5.27" rather than "127.10.5.*."

Note: There is no implied "*" on the beginning of host names. Thus, if you list "mypartner.com" on the Allowed Hosts list and nothing on the Denied Hosts list, then host ?? "is.mypartner.com"?? will not be allowed to connect.

Best practice: If you use a SAP BC server in your DMZ and one in your Intranet, the DMZ server should, if feasible, be configured with the Allowed Hosts list for all your trading partners, while the SAP BC server in the Intranet should include *only* the DMZ SAP BC server on its Allowed Hosts list.

Note: IP address and domain limitations apply only to incoming connections. They will not prevent your SAP BC server from initiating a connection to hosts not included in the list.

Best practice: If the set of organizations you interact with is relatively static, use IP address and/or domain limitations in your firewalls, filtering routers, *and* your SAP BC server. While the checks may be redundant, they provide added protection in the event that the firewall is breached.

In many organizations, protection by either domain name or IP address is not a usable scheme because the IP addresses and domain names change too frequently.

Best practice: IP addresses tend to change more than domain names, so it is more convenient to list domain names. However, it is safer to use IP addresses, since they are much harder to spoof than domain names. The tradeoff will depend on your willingness to make changes compared to your aversion to risk. If possible, configure your firewalls to detect (and reject) domain name spoofing attempts.

Note: IP address spoofing and DNS name spoofing can limit the value of these techniques, so you should not count on them as exclusive protections. They can, however, add significant value.

Note: IP address and domain-based restrictions control only who can and cannot connect, not what they can do once they have connected. Even if you use Allowed Hosts or Denied Hosts lists, you must still control access to the specific services (see below).

Understanding Users, Groups, ACLs, Folders, Services, and DSPs

The SAP BC server can control access to services, DSPs, and aliases. This section describes the methods used for each type of control. These concepts are described more fully in the *SAP BC server Administrator's Guide*; they are summarized here as a reminder of the concepts used in access controls. There are no best practices in this section.

Controls for Services

Figure 2 shows the relationships between users, groups, ACLs, folders, and services, and ports in SAP BC server.

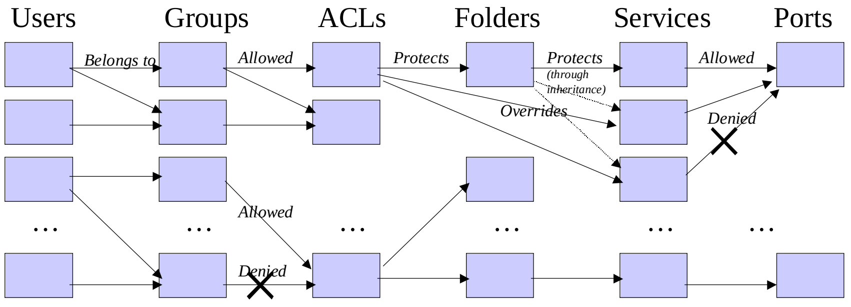


Figure 2. Relationship of Users, Groups, ACLs, Folders, Services, and Ports

- *Users* of the system represent partner organizations (for example, companies), as well as administrators and developers of the SAP BC server itself.

- Users belong to *groups*. A user can belong to an arbitrary number of groups; groups can have arbitrary numbers of members. There are certain built-in groups, such as Administrators, Anonymous**,** Developers, Replicators, and Everybody. You should create other groups to represent logical groupings of your users.

- An *ACL* is made up of two group listings: one listing of groups that are allowed access, and a second listing of groups that are denied access. Note that this is not allowing or denying access to the ACL itself, but rather will be used as part of the access control calculation when determining access to folders and services. Access is denied for ambiguous cases where the user is not a member of any groups listed (Allowed or Denied) or the user is a member of both a group in the Allowed list and a member of a group in the Denied list. The SAP BC server comes with several predefined ACLs: Administrators, Anonymous, Developers, Internal, Replicators, and Default. You should create other ACLs to represent other types of access desired.

- *Folders* are collections of *services*. A folder may have an associated ACL, in which case the ACL protects the folder by restricting who can access the services in the folder. A folder can have no more than one ACL.

- *Services* may also be protected by ACLs. If a service does not have an associated ACL, then the ACL of its parent folder (defined recursively) protects the service through *inheritance*. If a service has an associated ACL, then that ACL is used to determine access, and the ACL of any parent folder is ignored (that is, the ACL of the service overrides the ACL of the folder). A service can have no more than one ACL. You may use this technique to have a service that is more or less protected than the parent folder. For example, a folder may provide general-purpose services, a few administrative services, and a few anonymously accessible services. By placing the Default ACL on the folder, most of the services will be protected by inheritance, while the administrative services that need more protection might have an Administrators ACL, and the anonymously accessible services might have an Anonymous ACL.

- A *port* may provide access to one or more services. It may be configured to deny all services except those allowed (called Deny by Default) or to allow all services except those denied (called Allow by Default).

Access to services is determined as follows (you do not need to read this list to understand the remaining instructions in this document):

- If the port does not allow the source IP address (as described above in “Controlling by IP Address or Domain”), then the server rejects the request.

- If the port is configured as Deny by Default and the requested service is *not* included in the list of services to be allowed, then the server rejects the request.

- If the port is configured as Allow by Default and the requested service is included in the list of services to be denied, then the server rejects the request.

Note: Services invoked by another service within the same SAP BC server are not subject to the port constraints.

- If the user sending the request has not been authenticated, then the server processes the request as the Default user.

Note: By default, services invoked by another service within the same SAP BC server are not subject to ACL checks. You can configure ACL checks for internal service invocations on a per-service basis.

- If a service does not have an ACL specified, the server uses its inherited ACL; if no parent folder has an ACL set, then the server uses the Default ACL.

- If the requested service has an ACL, then the server allows the request only if the user is a member of at least one group listed on the ACL’s Allowed groups list and is not a member of a group listed on the ACL’s Denied groups. The server denies the request in all other cases.

Controls for DSPs and Public Files

Your SAP BC server can provide access to files stored in the file system, much as a web server does. It also includes support for Dynamic Server Pages (DSPs). The following explanation covers *all* files stored in the `*webMethods*\IntegrationServer4\packages\*packagename*\pub` directories (where `*<sapbc>*` represents the root of the installed directory tree and `*packagename*` is the name of the particular package, such as WmRoot). For example, the index.html file stored in each directory along with DSP files is subject to the same access rules described here.

The protection scheme for DSPs and other files stored in their directories is almost identical to that for services, with two exceptions:

- There are no port-based controls to limit which DSPs (or other files) can be accessed through a particular port.

- Access to DSPs (and other files) must be controlled for each directory. There is no inheritance or override mechanism as there is with folders and services.

Figure 3 shows the relationships for DSP protection, except that ports always serve all DSPs (you cannot restrict access to a DSP to a particular port). However, any services that a DSP invokes *are* subject to the access limitations of the port where the DSP was accessed.

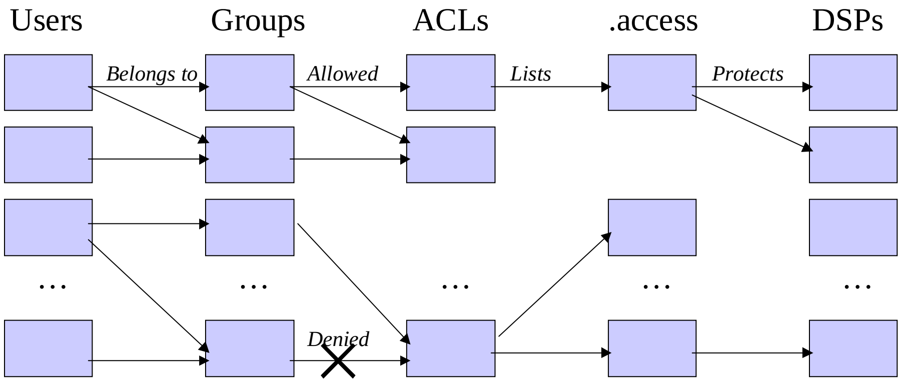


Figure 3. Relationship of Users, Groups, ACLs, .access Files and DSPs.

See "Controls for Services" on page 23 for a description of users, groups, and ACLs. Each directory that contains DSPs may have a `.access` file, either shipped with a SAP product or one that the administrator has created. If a `.access` file is present, it consists of lines, where each line gives the name of a DSP file in the directory (without the containing directory name) and the ACL associated with that DSP file. The `.`access file is always protected, so it does not require an entry for itself.

The following is an example of a `.access` file:

```
index.html Administrators
About.dsp Default
CheckPOStatus.dsp Partners
UpdatePOStatus.dsp Partners
```

Access to DSPs is determined as follows (you do not need read this list to follow the remaining instructions in this document):

- If the server does not allow the source IP address (as described above in "Controlling by IP Address or Domain"), then the server rejects the request.

- If the user sending the request has not been authenticated, then the server processes the request as the Default user.

- If the directory containing the DSP has a `.access` file and the `.access` file lists the requested DSP, then the server allows the request if the user is a member of at least one group listed on the ACL's Allowed group and is not a member of any group on the ACL's Denied groups list.
- If the directory containing the DSP does not have a `.access` file, or if it has an `.access` file that does not list the requested DSP, then the server applies the Default ACL.

Controls for Aliases

Aliases are used to access services on remote SAP BC servers. An alias consists of a name, a remote host name or IP address, a port, a remote user name and password, optional SSL settings, and an ACL. Aliases provide local users rights to use credentials registered on a remote server, and must therefore be carefully controlled.

IP or port access controls do not control access to aliases because they are only used by services on the local SAP BC server (for example, `pub.remote:invoke` and `pub.remote.gd:invoke`).

The server uses the assigned ACL to determine access to aliases. A user attempting to use an alias must be a member of at least one group on the ACL's Allowed list and not a member of any group on the ACL's Denied list.

By default, there are no aliases installed on the SAP BC server.

Setting Correct ACLs on Services

Follow this portion of the instructions for every SAP BC server.

As described above, it is important to have the proper ACLs on folders and services. Once you have completed setting up IP addresses (or domain names) that can connect to your SAP BC server, the next step in securing your server is to verify that every folder and service has the correct ACL. Folders and services may come from one of three sources:

- Folders and services that are built in to the SAP BC server.
- Folders and services that are custom built as part of your application, whether these services were built by SAP Professional Services, a SAP solutions provider, or your own organization.

Setting the correct ACLs for your server consists of four steps:

1. Verify that the factory settings for service/folder ACLs are unmodified for both built-in services and services provided by adapters and layered products.

2. Make a series of *recommended* changes to service/folder ACLs for both built-in services and services provided by adapters and layered products.
3. Consider a set of *suggested* changes to service/folder ACLs for both built-in services and services provided by adapters and layered products.
4. Review each of the custom services developed for your application, and assign appropriate ACLs.

The following sections describe each of these steps.

Verifying Factory Service/Folder ACL Settings

The first task in setting ACLs is to verify that the configuration of your SAP BC server does not have any changes that would weaken the security relative to the shipped settings.

Best practice: It’s best to check the ACL settings, even though it is time consuming.

Note: If you are confident that the ACLs for built-in services have not changed on your server since you installed it, you may skip this step.

To check the settings, review the ACL settings listed in “

Factory Folder/Service ACL Settings" for the services in the base SAP BC server, and make any changes to bring the ACL settings in line with the factory settings.

Also, check the factory settings for your adapters and layered products following the corresponding appendix of each such product. That is, if you use the SAP Adapter and the Oracle Adapter, you should follow the settings found in Appendix A of *both* the *SAP BC server SAP Adapter Security Best Practices* and *SAP BC server Oracle Adapter Security Best Practices*.

For instructions on how to set ACLs, see "Controlling Access to Services and Files with ACLs" in Chapter 8 of the *SAP BC server Administrator's Guide.*

Recommended Service/*Folder* ACL Setting Changes

The second task in setting ACLs is to make a series of improvements to the factory settings.

Best practice:"Recommended Folder/Service Changes to ACL Settings" lists a set of recommended ACL changes for the services in the base SAP BC server; we recommend that you make all of these changes. Also, check the corresponding appendixes for all adapters and layered products to see if there are recommended ACL changes for those products. That is, if you use the SAP Adapter and the Oracle Adapter, you should follow the settings found in Appendix B of *both* the *SAP BC server SAP Adapter Security Best Practices* and *SAP BC server Oracle Adapter Security Best Practices*.

For instructions on how to set ACLs, see "Controlling Access to Services and Files with ACLs" in Chapter 8 of the *SAP BC server Administrator's Guide.*

Best practice: SAP believes that the recommended ACL changes shown in "Recommended Folder/Service Changes to ACL Settings" will not affect the functioning of your SAP BC server. However, it is very important to test your application thoroughly before putting these changes into production.

Suggested Service/Folder ACL Settings Changes

The third task in setting ACLs is to consider making a set of ACL changes that *could* impact certain adapters and layered products or your own custom services. This set of changes requires investigation before making the modifications, as changing the ACLs could impact certain adapters and layered products or your own custom services.

"Suggested Changes to Folder/Service ACL Settings" lists a set of suggested ACL changes for the services in the base SAP BC server product. Also, check the corresponding appendixes for all adapters and layered products to see if there are suggested ACL changes for those products. That is, if you use the SAP Adapter and the Oracle Adapter, you should follow the settings found in Appendix C of *both* the *SAP BC server SAP Adapter Security Best Practices* and *SAP BC server Oracle Adapter Security Best Practices*.

Best practice: Carefully review the dependencies shown before making any of the changes. If you use any of the adapters or layered products shown in the "Dependency" column, you should not make the ACL change, as it could cause your application to malfunction.

For instructions on how to set ACLs, see "Controlling Access to Services and Files with ACLs" in Chapter 8 of the *SAP BC server Administrator's Guide*.

Best practice: After making any of the changes listed in "Suggested Changes to Folder/Service ACL Settings," it is very important to test your application thoroughly before putting these changes into production.

ACL Settings for Custom Services

In addition to the settings for the built-in services and the services provided by each of your adapters and layered products, you should review every service that is unique to your application, and set an appropriate ACL. The following general guidelines may be useful:

- Services that are used to modify the configuration of the application, or should otherwise be restricted to administrators of your SAP BC server, should generally have an Administrators ACL.
- Services that are only used during development of your application should generally have a Developers ACL. Most applications will not have any services in this category.
- Services that should be available to all authenticated users (regardless of who they are) should generally have the Default ACL.
- Services intended for use by your trading partners should have an ACL appropriate for the category. You may have a single group of trading partners, or you may have different sets with different levels of access to your services. If there are different sets of access, create separate groups (and separate ACLs).

You should avoid any services with an Anonymous ACL, as these will be accessible to any user who can connect to the port, even if they do not have a valid user ID (or client certificate). However the Anonymous ACL may be useful for selective, carefully controlled "guest" services.

Protecting Administrative Services

This portion of the instructions is optional.

As part of securing your SAP BC server, you need to protect the administrative services. You can do this in any of three ways (from most to least secure):

1. Use ACLs, coupled with a customized administrator port, to control access to administrative services.
2. Use ACLs alone without a specific port.
3. Rely on the firewall to protect an open port (a port for which all services are allowed).

Each of these methods has advantages and disadvantages, and are covered in the following sections.

Best practice: Using ACLs, coupled with a customized administrator port, is the best way of protecting administrative services on your server. However, this option requires the most work.

Customized Administrator-Only Port

The most secure way to protect administrative services is to identify the specific services that should be accessible to administrators, and to create a port for those services. The result is more secure, because even if an intruder gains access to the port, services with the Default ACL will not be available.

To use this option, create a new port with Deny by Default as the port type and add all services that are assigned the Administrators ACL. To create a new port, follow the instructions in “Configuring Ports” in Chapter 6 of the *SAP BC server Administrator’s Guide*. Next, follow the instructions in “Externally Visible Services” on page 60 to easily configure the port so it makes services that are appropriate for an administrator available. For additional information, refer to “Controlling Access to Services by Port” in Chapter 8 of the *SAP BC server Administrator’s Guide.*

Best practice: Even though the services are protected by the Administrators ACL, it is still a good idea to use your firewall or filtering router to protect this port against access by those outside your organization, as well as those inside the organization who do not need to administer the SAP BC server.

Note: Using this approach, you should avoid listing the administrative services on the externally visible services list for any port other than the port designated for the administrator’s use.

Using ACLs to Protect the Administrative Services

An easy alternative to setting up a separate administrator’s port is to rely on the ACLs as set up to protect the services, and not to set up a specific administrative port. This option brings moderate risk, because the administrative services are now exposed to any user who guesses the Administrator password or the password of any other user in the

Administrators group (subject only to the limits on IP addresses that can connect to your server).

To use this option, no additional configuration is necessary for either the SAP BC server or your firewalls. However, configuring the ACLs as defined in Appendixes A, B, and C is particularly critical in this case, since correct ACL configuration is the only protection provided.

Relying on the Firewall to Protect an Open Port

The simplest but least secure option for protecting administrative services is to create an Allow by Default port where all services are accessible, and then use your firewall or filtering router to protect against unauthorized access. This approach is not recommended, as it leaves your server vulnerable to any errors in firewall configuration.

To create a new port, follow the instructions in “Configuring Ports” in Chapter 6 of the *SAP BC server Administrator’s Guide*. Next, set the port type to Allow by Default, following the instructions in “Controlling Access to Services by Port” in Chapter 8 of the *SAP BC server Administrator’s Guide*. Do not list any services in the Deny list.

Caution: If you choose to have an open port, use firewalls and/or filtering routers to protect the open port against access both from outside users and unauthorized inside users.

Protecting Developer Services from Unauthorized Access

This portion of the instructions is optional.

Every SAP BC server requires developers to create the application. Generally, developers should not have access to the server once the application goes into production.

As part of securing your server, you need to control how developers use it. You can do this in any of the following ways (from most to least secure):

1. Use ACLs, coupled with a customized developer port, to control access to developer services.
2. Use ACLs alone without a specific port.
3. Rely on the firewall to protect an open port.

Each of these methods has advantages and disadvantages, as covered in the following sections.

Best practice: Using ACLs, coupled with a customized developer port, is the best way of protecting developer services on your server. However, configuring this option requires the most work.

Best practice: If you use ACLs and a customized developer port, or rely on the firewall to protect an open port, it is best to disable the developer port before moving into a production environment.

Developer supports the HTTP and HTTPS protocols (but not FTP or certificate-based authentication) for communicating with the SAP BC server. As a result, user name/password is the only form of authentication that developers can use on the port.

Customized Developer-Only Port

If you choose this option, you will create a new port that allows access to only those services used by developers, then use your firewalls to protect that port.

To create a new port, follow the instructions in "Configuring Ports" in Chapter 6 of the *SAP BC server Administrator's Guide*. Next, follow the instructions in "Externally Visible Services" on page 60 to easily configure the port so it makes services that are appropriate for a developer available. For additional information, refer to "Controlling Access to Services by Port" in Chapter 8 of the *SAP BC server Administrator's Guide*

Note: To perform application development, developers need access not only to the built-in services, but also to the services they develop. If possible, group developer services into one or more folders, so that you can give access using a wildcard. For example, if the application services can be grouped into the "acme" folder, then you can make the "acme" folder accessible through the developer port. By doing this, you will not have to list each individual folder and service used by the developer.

Best practice: Because some of your developed services will be intended for external use, and others for internal use only, it's best to group them into separate folders. For example, "acme.internal.*" could be the internal services, while "acme.external.*" could be the externally accessible services. You can give developers the ability to invoke "acme.*" on the developer port, while giving production users the ability to invoke only "acme.external.*" on the public port. You can do this even if internal and external services are mixed in the same folder, but it is easier if they are grouped separately.

Best practice: During the development process, be sure that the services being developed are protected by the Developers or the Internal ACL.

Best practice: Even though the services are protected by the Developers ACL, it is still a good idea to use your firewall or filtering router to protect this port against access by those outside your organization, as well as those inside the organization who do not need to administer the SAP BC server.

Best practice: If you use a customized developer-only port, you should avoid listing the developer services on the access list for any port other than the port designated for the developer's use.

Using ACLs to Protect the Developer Services

The second alternative is to rely on the ACLs as set up to protect the services, and not to set up a specific developer port. This is a simple alternative, but brings moderate risk, because the developer services are now exposed to any user who guesses the Developer password or the password of any other user in the Developers group (subject only to the limits on IP addresses that can connect to your server).

No additional configuration is necessary for either the SAP BC server or your firewalls. However, configuring the ACLs as defined in Appendixes A, B, and C is particularly critical in this case, since correct ACL configuration is the only protection provided.

Relying on the Firewall to Protect an Open Port

The simplest but least secure option for protecting developer services is to create an Allow by Default port where all services are accessible, and then use your firewall or filtering router to protect against unauthorized access. This approach is not recommended, as it leaves your server vulnerable to any errors in firewall configuration.

To create a new port, follow the instructions in “Configuring Ports” in Chapter 6 of the *SAP BC server Administrator’s Guide*. Next, set the port type to Allow by Default, following the instructions in “Controlling Access to Services by Port” in Chapter 8 of the *SAP BC server Administrator’s Guide*. Do not list any services in the Deny list.

Note: If you choose to use this option, you can use the same port for developers and administrators.

Best practice: If you choose to have an open port, use firewalls and/or filtering routers to protect the open port against access both from outside users and unauthorized inside users.

Protecting Replicator Services from Unauthorized Access

This portion of the instructions is optional.

You may use replication services to copy packages automatically from one SAP BC server to another. If you are sending or receiving packages from a trading partner, you may be using replication services.

As part of securing your SAP BC server, you need to control how replication will occur. This will depend in large part on whether you are replicating packages within your organization, or between organizations (for example, with partner servers). You can control access to replication services with the following methods:

1. Use ACLs, coupled with a customized replicator port, to control access to replication services.

2. Use ACLs alone without a specific port.

Each of these methods has advantages and disadvantages, as covered in the following sections.

Best practice: Using ACLs, coupled with a customized replicator port, is the best way of protecting replication services on your server. However, configuring this option requires the most work.

Best practice: HTTPS protects your packages from modification during transit, and is therefore the preferred transport protocol when performing replication.

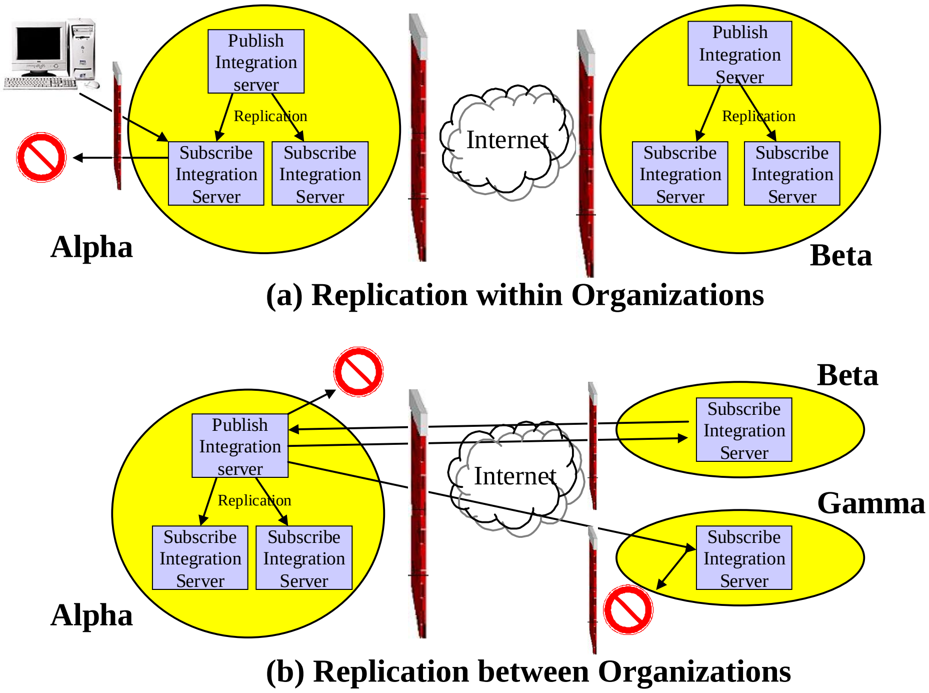
Replication with earlier SAP BC servers uses push technology (from a publishing server to subscribing servers). Beginning with SAP BC server 4.0, pull technology is also available (subscribing servers retrieving the from publishing server).

There are two parts to package replication:

1. **Setting up the Subscription.** This part is symmetric and, if you allow, both the publisher and the subscriber can set up the subscription.

2. **Package Distribution.** This is asymmetric and you can choose whether the Publisher can "push" packages when it chooses or whether the subscriber "pulls" packages when made available by the publisher.

Figure 4 shows replication within and across organizations.


Publish Integration server
Replication
Subscribe Integration Server
Subscribe Integration Server
Internet
Publish Integration Server
Replication
Subscribe Integration Server
Subscribe Integration Server
Alpha
Beta

(a) Replication within Organizations

Beta
Publish Integration server
Replication
Subscribe Integration Server
Subscribe Integration Server
Internet
Subscribe Integration Server
Gamma
Subscribe Integration Server
Alpha

(b) Replication between Organizations

Figure 4. Package Replication

In Part (a), the Alpha and Beta organizations each perform replication internally, but do not share services with each other. Attempts by users to connect to the replicator port (as described below) may be blocked by a firewall based on IP address. If the user is allowed through the firewall (or if there is no firewall), the user may be unable to perform any replication services without authenticating as a user who is a member of a group is assigned to the Replicators ACL. In this case, the publishing SAP BC server does not have a replicator port at all, because replication is configured for “push” operation only (that is, from the publishing server to the subscribing servers).

In part (b), the Alpha organization distributes its packages to the Beta organization. Alpha’s firewall must be configured to allow its SAP BC server to initiate a connection to Beta’s SAP BC server. Beta’s firewall must be configured to allow its SAP BC server to receive the connection on its replicators port. However, Beta’s firewall should not be configured to receive connections on its replicator port from Gamma’s SAP BC server. Gamma’s firewall is also configured to prevent connections from Alpha’s SAP BC server. Because replication is asymmetric, Alpha’s firewall may or may not allow through connections from Beta’s and Gamma’s SAP BC servers. If Alpha’s firewall blocks connections from Beta and Gamma, then Alpha’s administrator must perform all of the configuration on the Alpha SAP BC server. However, if Alpha’s firewall allows incoming connections from Beta and Gamma, they can sign up directly for replication services. For large trading networks, this may be more appropriate, since it imposes less of a burden on Alpha’s administrator.

In both scenarios, however, you may configure the SAP BC servers to exchange various types of documents. The limitation applies only to package replication.

General Replication Configuration

Before setting up a replicator port, you must configure the users and passwords to be used for replication. Throughout this discussion, the server on which the package originates is referred to as the *publisher*, and the recipient servers are the *subscribers*.

Setting up Replication for SAP BC server

The user name and password can be different on the publisher and the subscriber. The user name is not hard coded to be Replicator. The only restriction is that the user name used for replication must belong to the Replicators group or any other group that is assigned to the Replicators ACL. You *can* use SSL to set up the subscription and to distribute the package.

Note*:* The options described here are valid if *all* servers in the replication set (that is, all publishers and all subscribers) are running SAP BC 3.5.1 or higher. If you are running an earlier version, see the *SAP BC 3.x Security Best Practices*.

Publisher subscription setup: The publishing SAP BC server needs a user name and password on the subscribing SAP BC server. The user name must belong to the Replicators group or any other group that is assigned to the Replicators ACL. The subscriber will have to provide this user name to the publisher. In this use case, the subscriber does not need to have information about a user name/password on the publisher's end.

Subscriber subscription setup: The subscribing SAP BC server must have a valid user name and password for the publishing SAP BC server, and also needs to provide a user name and password to the publishing server. The publishing server will use this name when it is ready to distribute the package to the subscribing server.

Best practice: Delete the Replicator user from the SAP BC server. In its place, create a new user that belongs to the Replicators group. This user name can be different on the publisher and subscriber SAP BC servers. Using a name other than Replicator makes it harder for an attacker to guess the user name.

Best practice: Set the password for the newly created replication account to something that an attacker could not guess.

If the administrator for the publishing server does not want subscribers to have the ability to set up their own subscriptions and wants to manually set up subscriptions for all subscribers, then it is not necessary to provide a user name and password with replication privilege to all subscribers. In this case, the administrator of the publishing server can receive subscription requests via
e-mail or phone and enter these servers as subscribers.

Best practice: Use HTTPS (that is, an SSL-protected connection) to set up subscriptions and distribute the packages. This provides protection against theft of the software, as well as against tampering with the software in transit between the publishing and subscribing servers.

Continue with the section entitled "Customized Replicator-Only Port" or "Using ACLs to Protect the Replicator Services" (below).

Customized Replicator-Only Port

If you choose this option, you will create a new port that allows access to only those services used by replication, then use your firewalls to protect that port. The firewall protection may include allowing access to the port from your trading partners' SAP BC servers.

To create a new port, follow the instructions in "Configuring Ports" in Chapter 6 of the *SAP BC server Administrator's Guide*. Next, follow the instructions in "Externally Visible Services" on page 60 to easily configure the port so it makes services that are appropriate for a developer available. For additional information, refer to "Controlling Access to Services by Port" in Chapter 8 of the *SAP BC server Administrator's Guide*

Note*:* Using this approach, you should avoid listing the replicator services on the access list for any port other than the port designated for replication use.

Using ACLs to Protect the Replicator Services

The second alternative is to rely on the ACLs as set up to protect the services, and not to set up a specific developer port. This is a simple alternative, but brings moderate risk, because the replicator services are now exposed to any user who guesses the Replicator password or the password of any other user in the Replicators group (subject only to the limits on IP addresses that can connect to your server).

No additional configuration is necessary for either the SAP BC server or your firewalls to use this option. However, configuring the ACLs as defined in Appendixes A, B, and C is particularly critical in this case, since correct ACL configuration is the only protection provided.

Protecting Internal Services from External Access (Port Controls)

Follow this portion of the instructions for every SAP BC server.

The next step in securing your SAP BC server is to control access to those services needed by external users (in contrast to those services the server uses for internal functions). For example, external users need access to high-level database functions, but should not have access to services that access the database directly. The mechanism that controls which services are externally visible is known as *port controls*.

Best practice: To maximize security, we recommend that you identify those services that *should* be externally visible, and protect all other services. This approach is preferable to starting with a complete list of services and eliminating those that should be protected, since even one missed service could introduce security risks and endanger your server.

In addition to the services expressly used by Administrators, Developers, and Replicators (as described in previous sections), the following sets of services should be externally visible:

Client services: Services built in to the SAP BC server that must be accessible to use the SAP BCClient API. The server automatically adds these services to the allowed list when you create the port. These services are `wm.server:ping`, `wm.server:noop`, `wm.server:getServerNodes`, `wm.server:connect`, `wm.server:disconnect`

Guaranteed Delivery services: Services that are used for the SAP BCGuaranteed Delivery API The server automatically adds these services to the allowed list when you create the port.

These services are `wm.server.tx:start, wm.server.tx:restart, wm.server.tx:execute, wm.server.tx:end`. If you are not using the Guaranteed Delivery API, you do not need these services.

Adapter services: Services that are exported directly by the adapters you are using. See Appendix F of the *Security Best Practices* documents for your adapters, where you will find a list of services that must be exported for each adapter. You must enter these manually in the server administrator using the Access Edit Mode screen for the appropriate port..

Application-specific services: Services defined by your application that should be externally accessible. Obtain this list from the developers of your SAP BC server application.

Before proceeding, be sure you have a complete list of services that should be externally visible. The following sequence will set up your server to prevent access to all except the listed externally visible services. For details about how to perform each step, see “Controlling Access to Services by Port” in Chapter 8 of the *SAP BC server Administrator’s Guide*.

Best practice: Repeat these steps for each port *except* the administrative, developer, and replicator ports (set up in the previous sections).

1. Set the access mode for the port to Deny by Default, which is the default for new ports.

2. Add the adapter and Application-specific services that are to be available on the port.

Note*:* You might want to make some of your adapter services and application-specific services available from only particular ports. For instance, if you have services that only your preferred partners should access, make them available from port 5557 and use your firewall configuration to allow access to only the IP addresses of your preferred partners. In contrast, you could make all your publicly available services available from port 5555.

Note*:* When you set up the port controls, you will receive a warning that you may be preventing access to certain services. You may ignore this warning, provided that you have identified all services that should be externally visible.

Caution: When entering services or folders for a port, service/folder names should have the form `folder1.folder2.folder3:service`, where `folder1`, `folder2`, and `folder3` are the nested folders and `service` is the actual service name. Carefully check all values in the externally visible services list to ensure that you have entered them correctly.

Note*:* Make sure that you have selected the port type before proceeding with adding the service/folder names. Changing the port's access mode resets the ports service list to the default values.

Best practice: After setting up the externally available services, review the configuration for every port to verify that you have listed all services they should provide, and that the port type is set to Deny by Default.

Best practice: If there are some ports that cannot be set to Deny by Default, verify that your firewalls block access to those ports. If possible, block access to those ports from the inside too, allowing through only those individuals (such as administrators) who will need to access the Allow by Default port.

Relationship of Dynamic Server Pages and Port Controls for Services

Your SAP BC server can provide access to dynamic server pages (DSPs). Although access to the DSPs themselves is not limited by port controls, embedded service invocations in the DSPs *are* subject to port controls.

A DSP can be pure HTML, or more commonly, it includes service invocations of the form `%invoke pub.db:getTableInfo%`. If your DSPs include service invocations, each of the top-level services they invoke (in this example, (`pub.db:getTableInfo`) must be accessible through the port where the DSP is retrieved.

For example, the `db-alias.dsp` DSP directly invokes the following services:

```
wm.server.db:dataSourceList,
wm.server.db:dataSourceAdd
wm.server.db:dataSourceChange
wm.server.db:dataSourceDelete
```

For the operation to run successfully, the user that runs the db-alias.dsp DSP must be able to access each of these four services on that port. For a Deny by Default port (the recommended setting), each of the four services must be on the Folder/Service List. For an Allow by Default port, none of these services (or any of the folders that contain them) should be on the Folder/Service List.

If any of these four services invokes other services, the lower level services do *not* need to be externally visible on the port.

Setting Up .access Files

Follow this portion of the instructions for every SAP BC server.

Your SAP BC server can provide access to DSPs and other files. Control of DSP files is achieved via `.access` files placed in each directory. The controls you specify via .access files apply to *all* files in the `*webMethods*\IntegrationServer4\packages\*packagename*\pub` directory, whether the file is a DSP file, an HTML file, or some other type of file.

➡ **Note: Requests for the .access file are always denied. You do not need to include an entry for it in the `.access` file for itself.**

Unauthenticated users do not require access to DSPs to use the basic SAP BC server product or any webMethods-supplied adapters. Your environment may want to make certain files available this way, such as if you provide a "welcome" message telling partners how to sign up to perform online transactions.

Best practice: "Suggested Changes to Folder/Service ACL Settings" This appendix lists `suggested` changes to the ACL settings for folders and services to enforce stronger security than the default factory setting.

There are no additional suggested settings. Please refer to the "Recommended Folder/Service Changes to ACL Settings". These settings already provide a high level of security.

Best practice: Before you transition a system from testing and developing to production, you should identify all services used by your partners and replace the **Developers ACL** for those services by a **Partner ACL** containing the **Partner Group** in the allowed groups listing.

Factory .access file ACL SettingsThe Appendix "Factory .access file ACL Settings" lists the factory settings for `.access` files in the base product. You should check that all of the settings in your SAP BC server are at least as restrictive as those shown. Also, verify the `.access` settings for each adapter and layered product against the settings in the corresponding *Best Practices* document. That is, if you use the SAP Adapter and the Oracle Adapter, you should follow the settings found in Appendix D of *both* the *SAP BC server SAP Adapter Security Best Practices* and *SAP BC server Oracle Adapter Security Best Practices*.

"Recommended Changes to .access file ACL Settings" lists recommended changes to the factory settings. Similarly, you should check the corresponding appendix in each adapter and layered product's *Best Practices* to see if there are suggested `.access` settings for those products.

In addition, you should review all DSPs created as part of your application to determine appropriate settings for `.access` files.

Note*:* If the directory containing a DSP does not have a `.access` file, or if it has an `.access` file that does not list the requested DSP, then the server applies the Default ACL.

Note*:* An alternative to listing each file in the `.access` file is to create a `.access` file with a wildcard, for example:

```
* Administrators
```

This is a catchall that will protect all files not otherwise associated with an ACL. Attempts to retrieve the `.access` file are always denied.

Best practice: Using a text editor, create or edit the `.access` file for each directory under `pub.` Be sure that every DSP present in the directory is listed in the `.access` file with the appropriate ACL, as shown in the appendixes. Verify that there is a `.access` file for every directory, and every file in that directory has an appropriate ACL listed in the `.access` file.

Best practice: Do *not* use the Anonymous ACL unless you want to make the DSPs available to unauthenticated users.

Best practice: Protect the `.access` files from modification using your operating system's file protections. For example, on UNIX systems, the user ID running the SAP BC server should have read permission, but not write permission to the file. (If the SAP BC server is running as root, this will not be possible, since root has permission to write to every file on the system.)

Best practice: If your SAP BC servers is used in a production environments, the entire `pub` directory should be configured as read only to the user ID running SAP BC server.

Caution: If your server is running on a UNIX-based system, the listings for DSP files must be identical to the file names, including matching case. For example, if your `.access` file lists `CreateService.dsp`, but the actual name of the file is `createService.dsp`, the `.access` file entry will not provide the expected protection. If your server is running on a Windows system, case is unimportant, but it is still a good idea to match case exactly so that a future migration to a UNIX-based system would not expose your DSP files.

Just as with a web server, if a user requests access to a directory, the SAP BC server will return the contents of the `index.html` file (if one exists), or a listing of files in the directory (if there is no `index.html` file).

Best practice: Create an `index.html` file for every `pub` directory, or as an alternative, set the `watt.server.displayDirectories` parameter to "false" so that an attacker cannot access the list of files through your SAP BC server. Any attempt to access the directory will result in a "404" error (file not found).

➡ **Note***:* This parameter does not provide any protection against an attacker accessing the files themselves. It only prevents the attacker from obtaining a list of files available to be retrieved.

Best practice: Changes to `.access` files do not take effect until the containing package is reloaded. In the case of the WMRoot package, the changes do not take effect until you restart the SAP BC Server.

Chapter 8: Transitioning from Testing to Production

In many environments, looser controls are enforced when a system is in a testing environment compared to a production environment. Before you transition a system, it is important to verify that you have followed all recommendations in this *Best Practices* document for the production server, including:

- Setting the passwords for all privileged accounts
- Changing from password-based authentication to client certificate based authentication, if feasible
- Setting port access lists and ACLs appropriately
- Creating users with the minimum privilege possible, and verifying that non-administrative users are not members of the Administrators or Developers groups
- Verifying that you have installed all operating system and JVM patches
- Verifying that firewalls provide adequate protection from attacks
- Verifying that all ports are set to Deny by Default so non-listed services will be inaccessible. If this is not feasible, verify that your firewall prevents access to the ports that are set to Allow by Default.
- Verifying that replication services are protected from outsiders, both by verifying membership in the Replicators

group and by checking access to the replicators-only port (if you created one)

Best practice: Do not use the same password for the Administrator account on your production systems as you do on your development systems.

Best practice: SAP recommends disabling all developer access to the production system, both to prevent accidental changes to the configuration and to avoid providing attackers with a means of gaining privileged access to the SAP BC server. Remove all groups from the list of Allowed Groups on the Developers ACL. If this is not feasible, you should remove as many users as possible from the Developers group.

If you set up a developers-only port, consider disabling that port before going into production.

Best practice: The WmSamples package contains sample services that may be useful in a development environment. In a production environment, SAP recommends using the Server Administrator to either disable or delete this package.

Chapter 9: Auditing and Forensics

SAP BC server can keep extensive audit logs that can be useful in determining what happened in case of attack. Chapter 6 of the *SAP BC server Administrator's Guide* describes how to configure audit logs (see the section entitled "Working with Log Files"). Appendix C of the *SAP BC server Administrator's Guide* describes the contents of the logs.

Best practice: Audit logs are only useful if you know they have not been lost or tampered with. To maximize their value, it's best to store them off line by backing them up to tape or CD-R/RW media.

You may want to redirect your log files to a database or to an operating system log file (such as the *syslog* file on UNIX systems). You can use a custom event handler to redirect the SAP BC server's audit log. For details, see "Subscribing to Events" in the *SAP BC Developer Guide*.

Note: Be sure to back up log files regularly, and periodically move older log files off your system so that your file system does not fill up.

If an attacker were to gain access to your system, the integrity of your software and/or configuration files could be at risk. Many organizations are unable to detect or recover from such changes.

Best practice: Before bringing an integration system into production, calculate cryptographic checksums of all software and configuration files that should not change. Do *not* include audit logs, password files, and other files that will change during normal operation. Keep the checksums off line (for example, on a floppy disk), along with a copy of the software used to calculate the checksums and compare their values.

There are both freeware and commercial products that can calculate and verify checksums. Tripwire (www.tripwire.com) is the best known product in this genre. Some security assessment tools such as Symantec's Intruder Alert (www.symantec.com) also include similar capabilities. Finally, the `md5sum` command (available as part of the GNU utilities at www.gnu.org) provides similar capabilities. Do *not* use the `sum` command built into many systems, because attackers can easily modify programs or files without changing the checksums it generates.

Best practice: Before bringing an integration system into production, make a full backup to tape or CD-R/RW media of all software and configuration files. Make a new backup every time the configuration changes. Doing so allows you to recover in case of a successful attack.

If a system is compromised, neither cryptographic checksums nor backups will help you know whether the system processed transactions incorrectly. If the attacker has not compromised the audit trails, they may help in tracking down the attack.

Best practice: Don't wait for an attack before you start creating and reviewing audit logs. Once the attack has occurred, it's too late to generate logs of what happened. Proactively generating and reviewing audit logs is the best way to prepare. Periodically verify that your SAP BC server is generating logs, that they have sufficient information to determine what is occurring on the server, and that you know how to interpret them.

Note: SAP has no knowledge that any of its customers' systems have been attacked successfully. This recommendation is based on the premise that if such an attack were to occur, it would be too late to begin monitoring.

Chapter 10: Propagating Changes Across SAP BC servers

The instructions in this document have thus far described how to make the changes for a single SAP BC server. Configuration information, such as users, groups, ACLs, and ACL settings, are not automatically duplicated.

You may deploy new servers either for production or testing by copying configuration files from one server to another. We recommend copying entire files rather than using the SAP Server Administrator to set each attribute, which could result in errors and a less secure environment.

1. Install the new server according to the standard installation procedure documented in the readme.txt file.

2. Copy the files shown in the table below from the server that is configured correctly to the server that you want to have an identical configuration. The directories are expressed relative to the home directory of the SAP BC server.

Purpose	File name
ACLs and their relationship to groups	.\config\acls.cnf
ACL settings on services & folders	.\config\aclmap_sm.cnf
Event types and event subscriptions	*.\config\eventcfg.bin*
Port and associated protocols	.\config\port.cnf .\packages\WmRoot\config\listeners.cnf
Behavior when redirecting web automation	~~.\config\redir.~~cnf
Partner server signon information	.\config\remote.cnf
Users, groups memberships, and passwords	.\config\users.cnf
Assorted configuration parameters	.\config\server.cnf

If you are also copying user packages, you should copy .\config\evencfg.bin, which contains information for event types and subscriptions. Some of the services in the packages you copy might subscribe to events and therefore require this information.

In addition, you should copy all `.access` files that have been changed to the destination server (following the instructions in “Setting Up .access Files” on page 40).

Chapter 11: Using Tokens to Counter XSRF Attacks

Overview

Web browsers, or more precisely the servers they are connected to, are inherently prone to a security breach referred to as *cross-site request forgery* (XSRF). This vulnerability is a direct consequence of a usability requirement concerning authentication. If a server (or a web site) requires authentication, the logon data needs to be given only once – as opposed to each time a request is submitted to the server. For any subsequent requests directed at that server the credentials are provided by the browser. While this property of browsers is certainly welcome from a usability perspective (no user would appreciate a web site that keeps on asking for authentication – the same authentication – over and over again), it opens the door to unauthorized access, since a browser will add credentials to requests regardless of the true origin of those requests.

As an example, assume a user U is logged on to a site A secured by user and password. If the user opens another web page B while still logged on to A, there is a chance that B may contain malicious code in the shape of Javascript or hidden links (the `src` attribute of an image, for instance) that triggers a request to site A. That request will be properly authorized thanks to the browser and will be treated like a genuine request issued by user U. Thus the attacker has furtively gained access to site A and can possibly do serious damage and cause disruption as an imposter in the guise of a legitimate user U.

Browsers do not (yet) offer any safeguards against this type of attack. A user, on the other hand, is able to avert XSRF attacks by refraining from visiting any other sites while working with a secure site. This solution, however, is not acceptable since it imposes a very restrictive browsing behavior which is very likely to be (inadvertently) violated. For this reason BC 4.8 introduces the token concept.

Basics of Token Usage

A token is a random string associated with a particular user session upon successful logon. If token usage is enabled (details are given below) each request sent to the BC server via the administration UI should include the token pertaining to the respective session. If the token is missing, the BC server will reject the request and deny any kind of access (with the exception of so-called entry pages – see below).

The token is added to the request as a standard URL parameter in the form of a name/value pair, explicitly listed in links or added to forms as hidden input fields. As such, the token is not controlled by the browser and is therefore in particular not automatically added to requests issued in the context of XSRF attempts. As a consequence XSRF attacks are foiled.

Handling tokens without utilizing browser functionality comes at a cost: HTML (templates) or DSP pages need to be fitted with tokens at the appropriate places. Any request that can be sent to the originating server – and only those requests – must include the token. Manually adapting all pages is a very tedious approach that is prone to errors mainly through omission. Therefore the BC automatically enhances HTML and DSP pages to include the token. The automated approach, however, does not cover all contingencies. While it takes care of standard links that are listed in the DOM under `document.links` (defined through `<a href=”..”>`, for instance) and forms, it cannot handle Javascript statements like `location.replace(..)`, `window.location.href=..`, redirections via

```
<meta http-equiv="refresh" content="...;url=...">
```

or frame tags (in particular their `src` attribute). In those situations manual intervention is inevitable. (Details follow.)

The danger of XSRF attacks is relevant only for the administration UI. B2B scenarios are typically not threatened, mainly because there is no browser activity involved. Therefore, there is no opportunity for an attacker to trick a user into submitting a phony request with malicious intent as outlined above.

Since both administrative tasks and B2B activities may share the execution of some services, it is difficult, if not impossible, for the server to safely determine with certainty what the nature of a request is to decide whether a token is required or not. Fitting all third party software communicating through the BC with tokens is neither practical nor sensible. Therefore the necessity of tokens is determined on the basis of ports. By using distinct ports for administrative tasks (or UI centric scenarios) and B2B communication, the administrative tasks can be protected through tokens whereas communication passing through ports designated for B2B processes is exempt from token protection. When following this approach it is pivotal to separate the users authorized to access protected and unprotected ports. (See details below.)

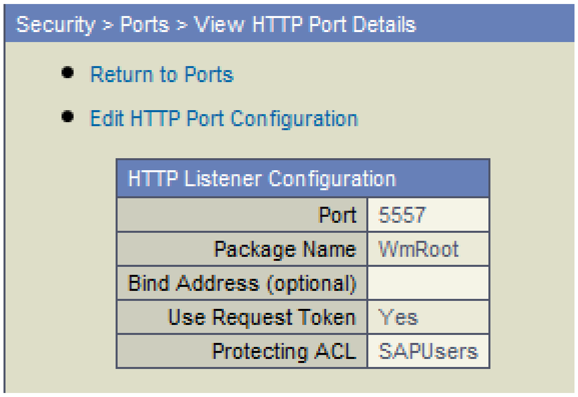
Enabling Token Usage

Token usage is enabled by setting

```
watt.server.session.enableRequestToken=true
```

in the server configuration file `server.cnf` or via Extended Settings. By default the value is `false` and hence token usage is generally switched off in order to not break existing scenarios.

When using tokens regular HTTP(S) ports and reverse invoke ports can be configured 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 presence or absence of the correct token. (A more detailed discussion of ACLs in the context of tokens is given in the section on pitfalls of token usage.)

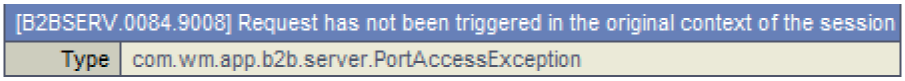


Security > Ports > View HTTP Port Details

- Return to Ports
- Edit HTTP Port Configuration

HTTP Listener Configuration	
Port	5557
Package Name	WmRoot
Bind Address (optional)	
Use Request Token	Yes
Protecting ACL	SAPUsers

If a port is configured to use tokens, any request sent through it that does not include a valid token will be rejected:



[B2BSERV.0084.9008] Request has not been triggered in the original context of the session	
Type	com.wm.app.b2b.server.PortAccessException

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 under Security. 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 entry pages with caution while keeping in mind that each entry page is a potential target for an attack that is never protected by a token.

As mentioned earlier, adding the appropriate token to a request is taken care of automatically except for pages that are not shipped with the BC installation and use certain tags and Javascript statements. The following section will explain how to manually adapt such pages.

One last remark with regard to reverse invoke: As there can be only a single configuration with regard to ACL and token, you should have separated setups for B2B and UI related scenarios.

Note: The SAP BC Developer cannot use a port protected with tokens. Hence, you should define a Developer only port protected by an ACL that only allows access for developers.

Manual Adaptation of DSPs

Manual adaptation of a DSP page is required, if the page contains requests directed at a token-protected port of a BC server by other means than standard links or forms. Let us assume a page contains the following Javascript statement, for instance:

```
location.replace("http://localhost:5556/Test/sample.dsp")
```

Let us further assume a BC server is running on `localhost` and 5556 is a token-protected port. Without modification execution of the statement will incur a port access exception as shown above.

The presence of a token can be tested with the DSP tag

```
    %ifvar -sapbcRequestToken%
```

If the DSP is called in a session that has an associated token, the tag will evaluate to true, and false otherwise.

The token (value) is available through

```
    %value $sapbcRequestTokenValue%
```

The name of the token to be used as the URL parameter name is

```
    %value $sapbcRequestTokenName%
```

Thus, the above Javascript statement needs to be extended as follows so as to make it work in a token-protected environment:

```
location.replace("http://localhost:5556/Test/sample.dsp
%ifvar -sapbcRequestToken%?%value
$sapbcRequestTokenName%=%value
$sapbcRequestTokenValue%%endif%")
```

Note that the token name can be configured through the server configuration property

```
watt.server.session.requestTokenQueryName
```

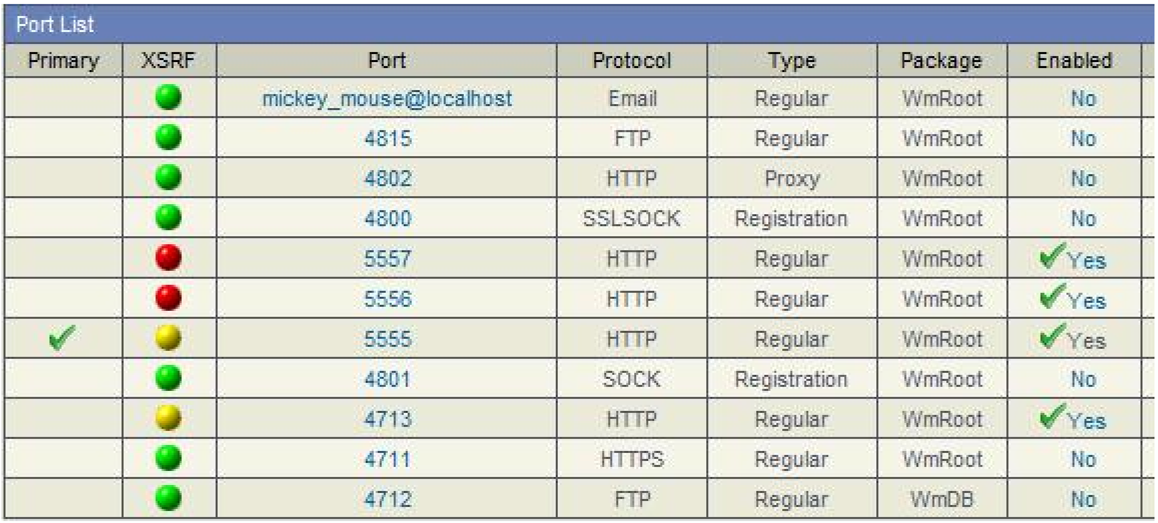
if for some reason the default value (`$sapbcRequestToken$`) needs to be changed.

Pitfalls of Token Usage

In the context of tokens every regular HTTP(S) port is protected through 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.



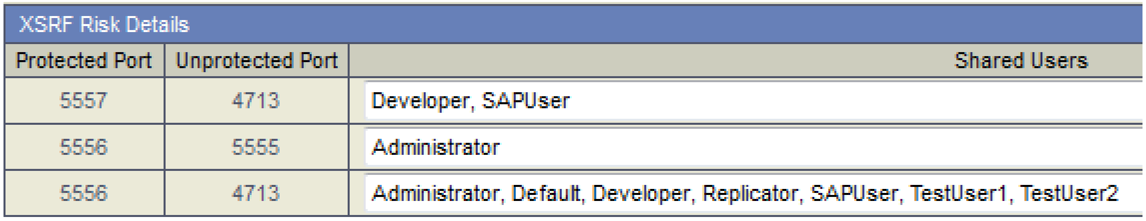
Port List

Primary	XSRF	Port	Protocol	Type	Package	Enabled
	🟢	mickey_mouse@localhost	Email	Regular	WmRoot	No
	🟢	4815	FTP	Regular	WmRoot	No
	🟢	4802	HTTP	Proxy	WmRoot	No
	🟢	4800	SSLSOCK	Registration	WmRoot	No
	🔴	5557	HTTP	Regular	WmRoot	✓Yes
	🔴	5556	HTTP	Regular	WmRoot	✓Yes
✓	🟡	5555	HTTP	Regular	WmRoot	✓Yes
	🟢	4801	SOCK	Registration	WmRoot	No
	🟡	4713	HTTP	Regular	WmRoot	✓Yes
	🟢	4711	HTTPS	Regular	WmRoot	No
	🟢	4712	FTP	Regular	WmDB	No

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.)



XSRF Risk Details

Protected Port	Unprotected Port	Shared Users
5557	4713	Developer, SAPUser
5556	5555	Administrator
5556	4713	Administrator, Default, Developer, Replicator, SAPUser, TestUser1, TestUser2

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 an XSRF attack. For port 5557, working with user Developer or SAPUser leaves port 4713 vulnerable to XSRF attacks.

Note that 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.

Chapter 12: Other Security Issues

In addition to the measures covered in this document, your security should include the following considerations:

- If your SAP BC servers or related systems are co-located (for example, operated by an outsourcing company), you should consider their security architecture. For example, do they use switches to create virtual LANs (VLANs)? If so, how is traffic protected to prevent information from flowing between VLANs? How are their firewalls configured? Do they provide adequate protection for your servers? In general, the co-location company will need to provide greater security than you would provide yourself, because they are likely to also be hosting servers operated by your competitors, and hence are more vulnerable to insider attacks.

- Physical security of your SAP BC server is critical. Apply the same protections to your server as any other mission-critical system in your environment to avoid theft, tampering, destruction, or water or electrical damage.

- Personnel security for your organization is critical. Because insider attacks make up a majority of security violations, consider how you verify and monitor employees. In particular, users with administrative rights (including anyone who is a member of the Administrators or Developers group) can do significant damage. Apply the same degree of personnel security as you do for administrators of other mission-critical systems.

- Denial of Service (DoS) attacks are virtually impossible to prevent. However, some firewalls can protect against certain types of attacks, such as those that open multiple connections to the SAP BC server. Check with your firewall vendor to determine the types of DoS attacks that it can stop.

Chapter 13: Building Secure Applications

The methods described in this document can help make your SAP BC server more secure. However, much of the security of your server depends on how carefully your build your application. This section provides a checklist of items to consider when designing, developing, testing, and deploying your SAP BC server application. It is not complete, but rather a starting point for consideration.

- **Do services provide the minimum capability possible?** For example, rather than providing a general service to query a database (and relying on the calling user to invoke it only in authorized ways), develop a more specific service that will allow only an *insert* or *query* operation. This gives more control, and reduces the possibility of a user using the service in an unintended manner.

- **Does the service check all inputs for validity?** Do not rely on a calling user to validate the input. For example, if you are passing input to a shell or a SQL query, verify that all possible special characters have been removed, so that a malicious caller cannot cause unexpected results. In general, it's better to allow known valid input, rather than trying to remove possible harmful input. That is, rather than trying to identify every character that could cause the underlying service to malfunction, if you know that a query should only contain letters, numbers, and spaces, allow *only* those characters instead of trying to remove all of the other characters.

- **Do you use "magic tokens," and if so are they easy to guess?** Many systems create a magic token, such as a cookie, and use that as authorization for a future action. If the token isn't truly random, then a malicious caller could guess the token, and thus gain access to another's authorization. Note that tokens based on the time of day are common, but are not sufficiently random, because an attacker can guess the time very accurately.

- **Do you execute external programs?** Executing an external program on behalf of a caller is potentially a dangerous operation, because it may allow the caller to force your host system to execute software that may have unintended results. Avoid executing external programs if possible. If you cannot avoid it, be sure to carefully validate all input against what the program is expecting, so a caller cannot cause unexpected results. In particular, avoid using a shell as an intermediary, because shells perform significant parameter interpretation.

- **Do you access web sites on behalf of a caller?** If so, the service can be used as a way to attack another site. Be sure that if you access another site, you validate the input.

- **Do you handle input beyond the range of expected values?** Will your service behave in an unexpected fashion if input is larger than you intended? For example, what will the service do if the name or identifier of an item on a purchase order is longer than expected? Rejecting the purchase order *may* be acceptable; overflowing a buffer must not occur because it is likely to lead to opportunities for bypassing the SAP BC server's security.

- **Do you have any "debugging" capabilities that bypass normal controls?** While debugging features are frequently helpful in the development process, they are frequently left in when a system enters production. Because such debugging capabilities frequently include back doors, it's best not to put them in at all, or to ensure that they are protected as any other service is.

- **Are you relying on encryption to solve all your security problems?** Encryption is a useful security feature, but it doesn't solve all security problems. Verify that your application is not relying on encryption to cover up weak solutions to security problems.

- **Are you using good encryption capabilities?** Almost anyone can come up with an encryption algorithm, but almost no one can come up with a *good* encryption algorithm. If you are using encryption for privacy, signatures, or any other purpose, use standard algorithms in standard ways. Proprietary encryption algorithms tend to not be as robust as standard ones.

- **Are you using different digital certificates for signing documents than for establishing SSL connections?** It is preferable to have two different pairs of digital certificates and corresponding private keys: one pair for establishing SSL connections and a second pair for signing documents.

- **Do you rely on obscurity for your system security?** If a system is truly secure, an attacker could have complete source code and still be unable to get in. Do not rely on an attacker being unaware of your design or implementation for protection.

Appendix A: Factory Folder/Service ACL Settings

This appendix lists the ACL settings for folders and services as released by SAP.

You can find instructions for using this appendix in "Verifying Factory Service/Folder ACL Settings" on page 28. If you are confident that you have not made any changes to the factory ACL settings, you may skip this section.

Each row in this table defines a single service or folder. The "Factory ACL Setting" column names the ACL that should be associated with the service. If you are confident that you have not made any changes to the factory ACL settings, you may skip this section.

⚠ **Caution:** For table entries that represent folders, verify that each folder has the recommended ACL, and that each folder or service within the folder has a folder that is not less restrictive. For example, if a folder has an ACL of Developer, it is safe to have a service within that folder with an ACL of Administrator. However, it would be unsafe to have a service with an ACL of Default, since that would include many users who are not members of the Developers group.

Service/Folder	Factory ACL Setting
WmDB	
pub	Internal
wm.server.db	Administrator
WmRoot	
wm	Internal
wm.dev:recording	Developers
wm.dev:util	Developers
wm.server.access	Administrators
wm.server.access:aclList	Developers
wm.server.access.adminui	Administrators
wm.server.admin	Administrators
wm.server.cache	Administrators
wm.server.cache.adminui	Administrators
wm.server.cache:resetCache	Developers
wm.server.codegen	Developers
wm.server:connect	Anonymous
wm.server:disconnect	Anonymous
wm.server.event	Developers
wm.server.event.getEventTypes	Developers
wm.server.flow	Developers
wm.server.flowGen	Developers
wm.server:getServerNodes	Anonymous
wm.server.jndi	Administrators
wm.server.jvm.threadDump	Internal
wm.server.jvm.threadDumpWithLocks	Internal
wm.server.ldap	Administrators
wm.server.net:changeIPAccessType	Administrators

Service/Folder	Factory ACL Setting
wm.server.net.ftp	Administrators
wm.server.net.http	Administrators
wm.server.net.https	Administrators
wm.server.net:ipRuleAdd	Administrators
wm.server.net:ipRuleDelete	Administrators
wm.server.net:ipRuleList	Administrators
wm.server.net.listeners	Administrators
wm.server.ns	Developers
wm.server:noop	Anonymous
wm.server.packages	Administrators
wm.server.packages.adminui	Administrators
wm.server.packages:getDependenciesList	Developers
wm.server.packages:packageActivate	Developers
wm.server.packages:packageCreate	Developers
wm.server.packages:packageDelete	Developers
wm.server.packages:packageInfo	Developers
wm.server.packages:packageReload	Developers
wm.server.packages:packageSettings	Developers
wm.server.packages:setPackageInfo	Developers
wm.server:ping	Anonymous
wm.server.portAccess	Administrators
wm.server.ports:listListeners	Administrators
wm.server.query	Administrators
wm.server.query.adminui	Administrators
wm.server.query:getSystemAttributes	Developers
wm.server.record	Developers
wm.server:reflect	Developers
wm.server.remote	Administrators
wm.server.remote:connectAndStartTx	Default
wm.server.remote:invoke	Default
wm.server.remote:invokeGD	Default
wm.server.remote:retrieve	Default
wm.server.replicator	Administrators
wm.server.replicator.adminui	Administrators
wm.server.replicator:getReleasedPkgInfo	Replicators
wm.server.replicator:packageSendZip	Replicators
wm.server.replicator:pullPackage	Replicators
wm.server.replicator:queryAvailablePackages	Replicators
wm.server.replicator:subscriptionAdd	Replicators
wm.server.replicator:queryPublisherForPackageInfo	Replicators
wm.server.replicator:queryPublisherForPackages	Replicators
wm.server.replicator:register	Replicators
wm.server.replicator:subscriberAdd	Replicators
wm.server.replicator:subscriberCancel	Replicators
wm.server.replicator:subscriptionCancel	Replicators
wm.server.replicator:unregister	Replicators
wm.server.sapjvm	Internal
wm.server.schedule	Administrators
wm.server.schema	Developers
wm.server.security	Administrators
wm.server.services	Administrators
wm.server.services.adminui	Administrators
wm.server.services:loadTemplate	Developers
wm.server.services:saveTemplate	Developers
wm.server.services:serviceInfo	Developers
wm.server.services:serviceInfoSet	Developers
wm.server.tx:end	Anonymous

Service/Folder	Factory ACL Setting
wm.server.tx:execute	Anonymous
wm.server.tx:init	Administrators
wm.server.tx:resetOutbound	Administrators
wm.server.tx:restart	Anonymous
wm.server.tx:sendTxErrorMail	Administrators
wm.server.tx:shutdown	Administrators
wm.server.tx:start	Anonymous
wm.server.ui	Administrators
wm.server.util	Internal
wm.server.util:getPackageFile	Replicators
wm.server.util:putPackageFile	Replicators
wm.server.util:remoteInvoke	Default
wm.server.web	Developers
wm.server.web:DocCacheClear	Default
wm.server.webtap	Developers
wm.server.xidl	Developers
wm.server.xidl.adminui	Administrators
WmPartners	
wm	Internal
wm.PartnerMgr.gateway	WmPartnersUsers
wm.PartnerMgr.gateway.admin	Administrators
wm.PartnerMgr.xtn	WmPartnersUsers
WmPublic	
pub	Internal
pub.cluster	Internal
pub.file:getFile	Internal
pub.webtap	Developers
WmSamples	
sample	Developers
tutorial	Developers
WmWin32	
pub	Internal
win32.COM	Developers
win32.ntlm	Administrators
SAP	Internal
sap.admin	Administrators
sap.transport	Administrators
sap.demo:handleRfcXMLPost	Administrators
sap.demo:handleIDocXMLPost	Administrators
sap.demo:handlebXMLPost	Administrators
sap.inbound	SAPUsers
sap.monitor:rfcTrace	Administrators
sap.rfc:createTemplate	Developers
pub.sap.transport.ALE:InboundProces	SAPUsers
pub.sap.transport.BAPI:InboundProcess	SAPUsers
pub.sap.transport.RFC:InboundProcess	SAPUsers
pub.sap.transport.XML:InboundProcess	SAPUsers

All inbound related services are protected by the ACL SAPUsers. The user under whose user id the SAP system connects to the BC SAP Adapter must have this ACL associated with it.´

The new user concept is described in greater detail in Chapter 10 of the *SAP Adapter Guide*.

Appendix B: Recommended Folder/Service Changes to ACL Settings Following Installation

This appendix lists *recommended* changes to the ACL settings for folders and services to enforce stronger security than the default factory setting. We strongly encourage you to make these changes as soon as is practical following installation.

Note: SAP believes that each of these ACL changes can be made without affecting your existing application. If your application uses any undocumented services or folders, some of these changes may interfere with your application. We urge you to perform a complete quality assurance test before making these changes to a production system.

Note: If your application uses any undocumented services or folders, some of these changes may interfere with your application. Always perform thorough testing before making changes to a production system.

Note: If you have installed further adapters also using the WmPartners Adapter, the recommended wm.PartnerMgr.xtn* settings might impact the proper function of these adapters. In this case you have to define a new ACL containing the SAPUsers group and all the users from your extra adapter needing access to the wm.PartnerMgr.xtn services in the allowed section.

You can find instructions for using this appendix in “Recommended Service/Folder ACL Setting Changes” on page 28.

Each row in this table defines a single service or folder. The “Recommended ACL Setting” column names the ACL that should be associated with the service.

Caution: For table entries that represent folders, verify that each folder has the recommended ACL, and that each folder or service within the folder has a folder that is not less restrictive. For example, if a folder has an ACL of Developer, it is safe to have a service within that folder with an ACL of Administrator. However, it would be unsafe to have a service with an ACL of Default, since that would include many users who are not members of the Developers group.

Service/Folder	Recommended ACL Setting
pub.event.WinException:eventLog	Administrators
win32.COM.dispatch:createObject	Developers
win32.COM.dispatch:invoke	Developers
win32.COM:invoke	Developers
win32.COM:invokeLate	Developers
win32.COM:shutdown	Developers
win32.ntlm:reg	Administrators
win32.ntlm:unreg	Administrators
sap.bapi.Browser	Administrators
sap.bapi:handleBusXMLPost	Administrators
wm.PartnerMgr.xtn	SAPUsers
wm.PartnerMgr.xtn:get	SAPUsers
wm.PartnerMgr.xtn:getAuditLog	SAPUsers

➡ **Note:** The win32 services are available only if your BC Server is running on Windows.

Appendix C: Suggested Changes to Folder/Service ACL Settings

This appendix lists suggested changes to the ACL settings for folders and services to enforce stronger security than the default factory setting.

There are no additional suggested settings. Please refer to the "Recommended Folder/Service Changes to ACL Settings". These settings already provide a high level of security.

Best practice: Before you transition a system from testing and developing to production, you should identify all services used by your partners and replace the **Developers ACL** for those services by a **Partner ACL** containing the **Partner Group** in the allowed groups listing.

Appendix D: Factory .access file ACL Settings

This appendix lists the ACL settings `.access` files as released by SAP.

For instructions for using this appendix, see "Recommended Service/Folder ACL Setting Changes" on page 28. If you are confident that you have not made any changes to the factory ACL settings, you can skip this section.

Each row in this table defines a DSP or other file, as specified by the package and DSP file name. The "Factory ACL Setting" column names the ACL that should be associated with the DSP file.

The name of the `.access` file is:

```
<sapbc>\server\packages\packagename\pub\.access
```

where *packagename* is the name from the first column of the table. The `.access` file should have one line for each DSP in that package. For example, for a SAP BC 4.0 Server, the contents of `<sapbc>\server\packages\WmDB\pub\.access` should be:

```
* Administrators
```

Note: Every `pub` directory must have its own `.access` file to list ACLs for the DSP and other files in that directory.

Package	DSP File	Factory ACL Setting
WmRoot	*	Administrators
	doc\*	Developers
	doc\OnlineHelp\*	Developers
	doc\OnlineHelp\images\*	Default
	icons\*	Anonymous
	images\*	Anonymous
	sap.css	Anonymous
	webMethods.css	Anonymous
	webMethods.js.txt	Anonymous
WmPublic	*	Default
WmWin32	*	Developers
WmDB	*	Administrators
	doc\*	Developers
	doc\OnlineHelp\*	Developers
	doc\OnlineHelp\images\*	Default
WmPartners	*	Administrators
	doc\*	Developers

Package	DSP File	Factory ACL Setting
	doc\OnlineHelp\*	Developers
	doc\OnlineHelp\images\*	Default
	images*	Default
	mailbox*	Default
WmSamples	*	Developers
	goes*	Developers
	mime*	Developers
SAP	Submit_IDocXML.html	Developers
SAP	*	Administrators
SAP	doc*	Developers
SAP	doc\OnlineHelp\*	Developers
SAP	doc\OnlineHelp\images\*	Default
SAP	icons*	Anonymous
SAP	images*	Anonymous

Appendix E: Recommended Changes to .access file ACL Settings

There are no suggested changes to the `.access` settings for SAP BC version 4.8.

Appendix F: Externally Visible Services

This appendix shows how to use the Server Administrator to easily create a list of services to be externally visible through a customized administrator, developer, or replicator port. Before performing this procedure, you must first create the customized port as described in “Customized Administrator-Only Port” on page 30, “Customized Developer-Only Port” on page 32 or “Customized Replicator-Only Port” on page 37.

Note: To determine the services that must be externally visible on the server for an adapter’s use, refer to Appendix F in the Best Practices document for that adapter. After obtaining the list, use the Edit Access

Mode screen of the Server Administrator to add the services to the allowed list to the appropriate ports. For additional information, refer to "Controlling Access to BC Services by Port" in Chapter 8 of the *SAP BC Administration Guide.*

Important: 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.

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.

   Use the pull down menu on the right of the screen to select the Administrators, Developers, or Replicators ACL. The server displays a list of the folders and services protected by this 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 the existing list. Then click Save Additions.
7. Click Done to return to the previous screen.

Important: The changes you make take effect immediately, even before you click **Done**. The function of the **Done** button is only to display the previous screen.

For additional information, refer to "Controlling Access to BC Services by Port" in Chapter 8 of the *SAP BC Administration Guide*

Best practice: Before you transition a system from testing and developing to production, you should identify all entry point services and make only these services accessible via a **Partner Port**. It is highly recommended that an HTTPS listener should be configured that denies all requests by default and that includes only these entry point services in the list of accessible services.

Best practice: SAP recommends disabling the **Developer Port** before transition to production.

For RFC/BC function maps from a SAP system to the BC server there exists no concept of entry point services. That means, all services that will be invoked directly or indirectly from an SAP system via an SAP listener at the BC server have to be accessible to a **Partner**.

Best practice: SAP recommends creating a **Partner SAPUsers Group** consisting of the SAP users (in capital letters) needing access to the BC server. The needed folders/services should be protected with the **SAPUsers ACL** containing the **Partner SAPUsers Group** in the allowed groups listing.