IBM Wave for z/VM
Version 1 Release 2

Administration and Customization

SC27-6118-14

Note:

Before you use this information and the product it supports, read the information in “Notices” on page 257.

This edition applies to Version 1.2 of IBM Wave for z/VM (product number 5648-AE1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2020-09-29

Contents

Figures

Tables

About this information

This document supports IBM® Wave for z/VM® (5648-AE1).

IBM Wave for z/VM is a provisioning and productivity management solution for simplifying the control and use of virtual Linux® servers and z/VM. IBM Wave is intended to significantly reduce the learning curve that is needed to manage and control z/VM and Linux guests. This information describes how you can maintain and customize IBM Wave for z/VM to meet the requirements for your environment.

Intended audience

This information is intended for Linux system administrators and z/VM administrators who are responsible for managing servers. IBM Wave for z/VM significantly reduces the learning curve needed to control the z/VM environment, which helps Linux and non-Linux z/VM system administrators continue to manage their servers with the skill set they currently possess.

Links to documents and websites

The PDF version of this information contains links to other documents and websites. A link from one PDF file to another PDF file works only when both files are in the same directory or database. Links to websites work when you have internet connectivity. A document link is to a specific edition. If a newer edition of the linked documents is published, ensure that you have the current edition.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

To send us your comments, go to z/VM Reader's Comment Form (www.ibm.com/systems/campaignmail/z/zvm/zvm-comments) and complete the form.

If you have a technical problem

Do not use the feedback method. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See IBM: z/VM Support Resources (www.ibm.com/vm/service).
- Go to IBM Support Portal (www.ibm.com/support/entry/portal/Overview).

Conventions and terminology

This topic includes some of the typographic conventions and terminology used in this publication.

Typographic conventions

The following table describes the typographic conventions used in this publication.

Table 1. Conventions

Convention	Usage
bold	Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options. Depending on the context, **bold** typeface sometimes represents path names, directories, or file names.
bold underlined	bold underlined keywords are defaults. These take effect if you do not specify a different keyword.
constant width	Examples and information that the system displays appear in constant-width typeface. Depending on the context, constant-width typeface sometimes represents path names, directories, or file names.
italic	*Italic* words or characters represent variable values that you must supply. *Italics* are also used for information unit titles, for the first use of a glossary term, and for general emphasis in text.
<key>	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <Enter> refers to the key on your terminal or workstation that is labeled with the word *Enter*.

****	In command examples, a backslash indicates that the command or coding example continues on the next line. For example:
	`mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \`
	`-E "PercentTotUsed < 85" -m p "FileSystem space used"`

{item}	Braces enclose a list from which you must choose an item in syntax descriptions.
[item]	Brackets enclose optional items in syntax descriptions.
<Ctrl-x>	The notation <Ctrl-*x*> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.
item...	Ellipses indicate that you can repeat the preceding item one or more times.
|	In syntax statements, vertical lines separate a list of choices. In other words, a vertical line means *Or*. In the left margin of the document, vertical lines indicate technical changes to the information.

Terminology

This section includes some of the common terms used in this publication.

Term
Description

Wave
A provisioning and productivity management solution for simplifying the control and use of virtual Linux servers and z/VM. Synonym for *IBM Wave*, *IBM Wave for z/VM*.

Wave client
The Wave user interface. Synonym for *Wave client application*, *Wave GUI*, *Wave GUI application*.

WAVESRV
One or more servers that comprise the second tier in Wave's three-tier architecture (see Figure 1 on page 3). Synonym for *Wave Linux server*, *WAVESRV server*.

Summary of changes for Wave 1.2: Administration and Customization

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Fix Pack 16 changes (September 2020)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 16.

New information

IBM Wave for z/VM recognizes and supports z/VM 7.2.

The following topics are new:

- “Testing the connection to SMAPI and the service machines” on page 69
- “Set IBM Wave Database Backup Password” on page 100

Updated information

The following topics are updated:

- “Prerequisites for z/VM” on page 51
- “Prerequisites for Linux guests” on page 52
- “Prerequisites for the IBM Wave Linux server” on page 53
- “Review the TCP/IP settings” on page 53
- “Configure SMAPI” on page 54
- “Authorize DirMaint” on page 55
- “Setting up Performance Toolkit for z/VM” on page 56
- “Authorizing the service machines with RACF” on page 60
- “Install the Wave Linux server (WAVESRV)” on page 62
- “Start IBM Wave for z/VM” on page 66
- “Running the Auto-Detect Wizard” on page 71
- “Adding, updating, and deleting Linux media repositories” on page 96
- “IBM Wave Database Actions” on page 99
- “Backup IBM Wave Database” on page 100
- “Thresholds and Defaults” on page 113
- “Enterprise Directory parameters” on page 127
- “IBM Wave security tasks” on page 137
- “IBM Wave user authentication” on page 143
- Appendix D, “Changing the IBM Wave server IP address or host name,” on page 171
- Appendix G, “Configuring VM: Secure,” on page 181
- “TCP/IP” on page 182
- “Generating a new certificate and signing it” on page 195

Fix Pack 15 changes (April 2020)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 15.

New information

IBM Wave for z/VM recognizes and supports:

- IBM z15 Model T02 (see “IBM mainframe requirements” on page 1)
- IBM LinuxONE Model LT2 (see “IBM mainframe requirements” on page 1)
- Red Hat Enterprise Linux 8 (RHEL 8) (see Appendix A, “Linux distribution support,” on page 165.

The following topics are new:

- “Upgrading IBM Wave for z/VM” on page 68
- “Starting and stopping the WebSphere Liberty server” on page 83.

Updated information

The following topics are updated:

- “The Wave Linux server (WAVESRV)” on page 8
- “The Background Task Scheduler (BTS) server” on page 9
- “FCP-attached storage” on page 28
- “Prerequisites for the IBM Wave Linux server” on page 53
- “Installing IBM Wave for z/VM” on page 62
- “Install the Wave Linux server (WAVESRV)” on page 62
- “Port reference information” on page 77
- Chapter 3, “Wave APIs and WebSphere Liberty,” on page 83
- “Wave embedded WebSphere Liberty” on page 83
- “External Entities Manager” on page 85
- “Manage Device Pools” on page 86
- “Manage Virtual Network Segments” on page 87
- “Changing the source directory” on page 89
- “Update Authorized TVP-API Credentials” on page 92
- “IBM Wave Database Actions” on page 99
- “Backup IBM Wave Database” on page 100
- “Regenerate IBM Wave Database Password” on page 101
- “Regenerate Encryption Keys” on page 101
- “Recycle Service Machines” on page 112
- “BTS parameters” on page 117
- “IBM Wave server options” on page 134
- “Other Wave server options” on page 136
- “IBM Wave security tasks” on page 137
- “Creating and updating IBM Wave user profiles” on page 155
- Appendix L, “Managing Wave's server certificate,” on page 195
- “Generating a new certificate and signing it” on page 195
- “Viewing the server certificate” on page 198.

Deleted information

- Support for installing the Wave server on Red Hat Enterprise Linux 6 (RHEL 6) is withdrawn. Wave continues to support managing RHEL 6 guests.
- Support for installing the Wave server on SUSE Linux Enterprise Server 11 (SLES 11) is withdrawn. Wave continues to support managing SLES 11 guests.

Fix Pack 14 changes (December 2019)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 14.

Updated information

The following topics are updated:

- “Install the Wave Linux server (WAVESRV)” on page 62
- “Running the Auto-Detect Wizard” on page 71
- “Update Authorized TVP-API Credentials” on page 92
- “Adding, updating, and deleting Linux media repositories” on page 96
- “Linux media repository creation processing” on page 97
- “Linux media repository update processing” on page 97
- “Linux media repository delete processing” on page 98
- “Functionality parameters” on page 119
- “Security parameters” on page 125
- “Creating and updating IBM Wave users” on page 151
- “Generating a new certificate and signing it” on page 195

Fix Pack 13 changes (September 2019)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 13.

New information

- IBM Wave for z/VM recognizes and supports IBM z15 and IBM LinuxONE III. (see “IBM mainframe requirements” on page 1)
- IBM Java™ 1.8 is required on the Wave server (see “Prerequisites for the IBM Wave Linux server” on page 53)
- Java 1.8 is required on workstations that run IBM Wave (see “Prerequisites for workstations that run IBM Wave” on page 53)
- “Conventions and terminology” on page xix
- “IBM Wave security tasks” on page 137
- “Disabling Wave server certificate validation in the IBM Wave client” on page 142
- “WAVEPasswordResetter command” on page 202
- New messages (in Appendix N, “IBM Wave messages,” on page 203):
  - HWVP0100I
  - HWVP5100E

Updated information

The following topics are updated:

- “Intended audience” on page xv
- “The Background Task Scheduler (BTS) server” on page 9
- “Mechanism” on page 26
- “Review the parameter files” on page 30
- “Inconsistency mechanism” on page 34
- “Authorize DirMaint” on page 55
- “Configuring IBM Wave service machines ” on page 57
- “Authorizing the service machines with RACF” on page 60
- “Installing IBM Wave for z/VM” on page 62
- “Install the Wave Linux server (WAVESRV)” on page 62
- “Start IBM Wave for z/VM” on page 66
- “Thresholds and Defaults” on page 113
- “GUI parameters” on page 116
- “BTS parameters” on page 117
- “Functionality parameters” on page 119
- “NFS parameters” on page 122
- “Attention Required Definitions” on page 124
- “Security parameters” on page 125
- “Enterprise Directory parameters” on page 127
- “Audit Log parameters” on page 129
- “The password resetter utility” on page 142
- “IBM Wave user authentication” on page 143
- Chapter 7, “User management,” on page 147
- “Understanding user types and roles” on page 147
- Appendix E, “Shared directory considerations for service machines,” on page 173
- Appendix F, “Considerations for the service machines when working with SSI,” on page 177
- Appendix J, “Configuring certificates for managed z/VM systems,” on page 191
- Appendix K, “Using SSL and TLS certificates for LDAP or Active Directory login,” on page 193

Updated messages (in Appendix N, “IBM Wave messages,” on page 203):

- HWVP0001I
- HWVP5001E

Deleted information

The following topic has been removed:

- From Chapter 1: "First log on for the IBM Wave user interface"

Fix Pack 12 changes (April 2019)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 12.

New information

- IBM Java 1.8 is required on the Wave server when running the IBM Wave APIs. See “Prerequisites for the IBM Wave Linux server” on page 53.
- “Firewall information” on page 80.

- “Changing a keystore password” on page 200.

Updated information

The following topics are updated:

- “The Wave Linux server (WAVESRV)” on page 8
- “Functionality and Activation Levels and Activation Done signaling” on page 21
- “Prerequisites for z/VM” on page 51
- “Install the Wave Linux server (WAVESRV)” on page 62
- “Start IBM Wave for z/VM” on page 66
- “Port reference information” on page 77
- Chapter 3, “Wave APIs and WebSphere Liberty,” on page 83
- “Wave embedded WebSphere Liberty” on page 83
- “Enterprise Directory parameters” on page 127
- “Wave server log options” on page 134
- “Tasks for uninstalling IBM Wave” on page 163
- Appendix D, “Changing the IBM Wave server IP address or host name,” on page 171
- Appendix L, “Managing Wave's server certificate,” on page 195
- “Generating a new certificate and signing it” on page 195
- “Converting a JKS keystore to PKCS12” on page 198.

Deleted information

- Support for Red Hat Enterprise Linux 5 (RHEL 5) is withdrawn.
- The `signalActivationDone` CLI command and related information have been removed.

Fix Pack 11 changes (September 2018)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 11.

New information

IBM Wave for z/VM recognizes and supports:

- z/VM 7.1
- SLES 15 (see Appendix A, “Linux distribution support,” on page 165.

A new appendix explains how to configure IBM Wave for zMON. For more information, see Appendix I, “Configuring IBM Wave for zMON,” on page 185.

Updated information

The following topics are updated:

- “Application architecture” on page 2
- “Supported target virtualization platforms (TVPs)” on page 3
- “The Background Task Scheduler (BTS) server” on page 9
- “The Wave Linux server (WAVESRV)” on page 8
- “BTS Live Guest Relocation sampler” on page 11
- “Guest logon eligibility” on page 13
- “Single system image and live guest relocation support” on page 15

- “Resource verification before activation” on page 21
- “Prototype management” on page 28
- “Directory manager generated work units” on page 36
- “z/VM account management” on page 36
- “Running Automatic Guest Classification (AGC)” on page 43
- “The Attention Required mechanism” on page 47
- Chapter 2, “Installing and customizing IBM Wave,” on page 51
- “Prerequisites for z/VM” on page 51
- “Prerequisites for Linux guests” on page 52
- “Prerequisites for the IBM Wave Linux server” on page 53
- “Prerequisites for workstations that run IBM Wave” on page 53
- “Review the TCP/IP settings” on page 53
- “Configure SMAPI” on page 54
- “Authorize DirMaint” on page 55
- “Configuring IBM Wave service machines ” on page 57
- “Creating the service machines” on page 57
- “Authorizing the service machines with RACF” on page 60
- “Authorizing the service machines in other ways” on page 62
- “Thresholds and Defaults” on page 113
- “GUI parameters” on page 116
- “BTS parameters” on page 117
- “Functionality parameters” on page 119
- “NFS parameters” on page 122
- “Attention Required Definitions” on page 124
- “Security parameters” on page 125
- “Enterprise Directory parameters” on page 127
- “Audit Log parameters” on page 129
- “Linux Login Security Options” on page 140
- “Displaying audit log events” on page 159
- “WaveConnectableGuestsExit - Connectable guests exit” on page 249
- Appendix J, “Configuring certificates for managed z/VM systems,” on page 191
- Appendix E, “Shared directory considerations for service machines,” on page 173
- Appendix F, “Considerations for the service machines when working with SSI,” on page 177
- Appendix K, “Using SSL and TLS certificates for LDAP or Active Directory login,” on page 193

Fix Pack 10 changes (May 2018)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 10.

New information

IBM Wave for z/VM recognizes and supports IBM z14® Model ZR1 and IBM LinuxONE Rockhopper II. See “IBM mainframe requirements” on page 1.

Fix Pack 9 changes (December 2017)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 9.

Updated information

The WAVESRV directory example is updated with larger storage sizes. For the example, see Appendix C, "A sample WAVESRV directory entry," on page 169.

Fix Pack 8 changes (October 2017)

This edition includes changes to support product changes provided for the general availability of Wave for z/VM 1.2 Fix Pack 8.

New information

- IBM Wave for z/VM recognizes and supports IBM z14 and IBM LinuxONE Emperor II. For more information, see "IBM mainframe requirements" on page 1.
- Workstations that run IBM Wave can now use Java 1.8 and Microsoft Windows 10. For more information, see "Prerequisites for workstations that run IBM Wave" on page 53.
- There is a new option available to prevent the auto-detect work unit from failing when IBM Wave does not recognize a system. For more information, see "Other Wave server options" on page 136.

Fix Pack 7 changes (March 2017)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 7.

New information

- IBM Wave provides a RESTful API server. For more information, see Chapter 3, "Wave APIs and WebSphere Liberty," on page 83.
- A new topic describes how IBM Wave represents dormant guests. For more information, see "Enterprise Directory parameters" on page 127.
- A new appendix explains how to set up SSL certificates that allow LDAP login over SSL. For more information, see Appendix K, "Using SSL and TLS certificates for LDAP or Active Directory login," on page 193.
- A new appendix explains how to set up SSL certificates that provide a more secure communication with the IBM Wave API server. For more information, see Appendix L, "Managing Wave's server certificate," on page 195.

Updated information

- The IBM Wave API uses a specific port for RESTful communication. For more information, see Table 9 on page 77.
- IBM Wave requires a `zip` package for installing the IBM Wave API. For more information, see "Prerequisites for the IBM Wave Linux server" on page 53.
- IBM Wave parameters now contain a new parameter that controls the way the IBM Wave API responds to requests involving objects that have IANs attached to them. For more information, see "Functionality parameters" on page 119.
- The IBM Wave API provides more logging options. For more information, see "Wave server log options" on page 134.
- The IBM Wave uninstall procedure now contains steps for uninstalling the embedded WebSphere Liberty server. For more information, see Chapter 9, "Uninstalling IBM Wave," on page 163.

- The IBM Wave Parameters include three new options that can control how many days work units are kept, based on the type of user who initiated them. For more information, see “BTS parameters” on page 117.
- The IBM Wave Login Parameters for LDAP over SSL were changed. IBM Wave supports a local keystore and allows changing the password that is used to access the certificates. For more information, see “Enterprise Directory parameters” on page 127.
- If VM:Secure code does not reside in the default location a LINK statement should be added to IBM Wave service machines directory entry. For more information, see “Service Machine” on page 182.
- The IBM Wave API server produces new messages. For more information, see Appendix N, “IBM Wave messages,” on page 203.

Fix Pack 6 changes (December 2016)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 6.

New information

- IBM Wave automatically detects when a user must be authenticated with the LDAP server or as an IBM Wave user. This feature removes the LDAP option from the Login screen and changes the parameter settings. For more information, see “Enterprise Directory parameters” on page 127.
- IBM Wave users can now be deleted. For more information, see “Deleting IBM Wave Users” on page 154
- IBM Wave can now add, auto-detect, and manage a z/VM 6.4 system.

Updated information

- IBM Wave requires `deltarpm` when installing on RHEL 7.2 and SLES12. For more information, see “Prerequisites for the IBM Wave Linux server” on page 53
- The topic that describes the configuration of SMAPI and DIRMAINT was rewritten. The new content contains all the needed information for each z/VM release. For more information, see “Configuring TCP/IP, SMAPI, and DirMaint” on page 53
- List of search paths for commands when running Linux commands on managed guest.

Fix Pack 5 changes (September 2016)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 5.

New information

- When you first start IBM Wave, after service pack 5 (SP5), a **What's New** pane displays the new function in IBM Wave. The pane also contains a dismiss option. To reopen the pane, select **Help** > **What's New**.
- You can now dynamically increase memory assigned to an active Linux guest (in addition to dynamically adding CPUs). For more information, see the following topics:
  - The **Manage Parameters** > **Threshold** contains new fields for controlling the memory values. For information, see “Thresholds and Defaults” on page 113.
  - The **z/VM Project Permissions** contains a new permission to "Add Memory Dynamically". For more information, see “Overview of scopes and permissions” on page 148. Note that during service pack 5 (SP5), to avoid any migration actions, the default permission to "Add Memory Dynamically" remains that same as the "Update" permission.
- You can now have the option of extending storage from a virtual group. For more information, see the “Functionality parameters” on page 119.

Fix Pack 4 changes (June 2016)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 4.

New information

- IBM Wave for z/VM now supports Ubuntu Server 16.04.
- You can now dynamically increase CPUs that are assigned to active Linux guests without recycling the guest. For more information, see the following topics:
  - The **Manage Parameters** > **Threshold** contains new fields for controlling the CPU values. For information, see "Thresholds and Defaults" on page 113.
  - The z/VM user permissions contains a new permission to "Add CPUs Dynamically". For more information, see "Overview of scopes and permissions" on page 148. Note that during service pack 4, to avoid any migration actions, the default permission for Adding CPU remains that same as the Update permission.
- The information about Automatic Guest Classification (AGC) is rewritten, clarified, and consolidated into the following topics:
  - "Automatic Guest Classification" on page 37
  - "AGC Manager" on page 37
  - "Defining AGC entries" on page 41
  - "Running Automatic Guest Classification (AGC)" on page 43
  - "Resolving AGC conflicts and inconsistencies" on page 44
- An NFS prerequisite for the command line interface (CLI) is added to "Installation prerequisites" on page 51.
- To run a script on a managed Linux guest, you must have an NFS client installed. For more information, see "Prerequisites for Linux guests" on page 52.
- IBM Wave for z/VM Service Pack 4 (SP4) includes port reference material. See "Port reference information" on page 77.
- The **Updating Minidisk Passwords** in the **Administrative** > **Site Management** menu is no longer erroneously missing from this information. For more information, see "Update Minidisk Passwords" on page 93.
- IBM Wave now issues a warning message when someone, other than the site/system level administrator, tries to activate a guest on a z/VM system that is not the default z/VM system. See the new option in "Functionality parameters" on page 119.
- A new topic is added to explain about Appendix D, "Changing the IBM Wave server IP address or host name," on page 171.

Updated information

- For Ubuntu Server support, see the following topics:
  - "The Wave Linux server (WAVESRV)" on page 8.
  - "Bare-metal installation (BMI) support" on page 25.
  - "Installation prerequisites" on page 51.
  - "IBM Wave Linux Repository Manager" on page 95.
  - "WAVECloneConfigExit - Cloned server first boot exit" on page 249.
  - Appendix A, "Linux distribution support," on page 165.
- Changes are made to accurately reflect the performance data that is retrieved through Performance Toolkit for z/VM. For more information, see "z/VM utilization and performance statistics" on page 34.

- The **Storage Viewer** includes enhancements to help you avoid mismatches with storage allocations and DASD volumes contain more information from z/VM and DIRMAINT. For information, see "DASD storage status" on page 27 and the "Attention Required Definitions" on page 124 parameters tab, which contains new rows for DASD volumes and DASD groups.
- When making NFS updates, you must enter your authentication credentials. For information, see the "NFS parameters" on page 122.
- For LDAP authentication, the CA Certificate field is clarified. The path provided must be on the Wave Server before the configuration is successfully saved. For more information, see "Enterprise Directory parameters" on page 127.
- Additional information about security for the *WAVEuser* is added to "Linux Login Security Options" on page 140.
- When using the **Program Parameters Syntax** in "Changing User Preferences" on page 132, a warning message is issued.
- If you use the "WaveConnectableGuestsExit - Connectable guests exit" on page 249 sample exit from, ensure the **nmap** is installed on the IBM Wave server.
- The XPRFEXIT SAMPEXEC sample file is not shipped with IBM Wave for z/VM. Instead, use the example provided in the following topic: "XPRFEXIT - PROFILE EXEC exit for service machines" on page 250.

Deleted information

IBM Wave for z/VM removed the requirements to add PRIVCLASS C and E to DIRMAINT, and class Z DIRMAINT command authority. The information was removed from "Authorize DirMaint" on page 55.

Fix Pack 3 changes (April 2016)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 3.

New information

- IBM Wave supports IBM z13s®, IBM LinuxONE Rockhopper, and IBM LinuxONE Emperor. For more information, see "IBM mainframe requirements" on page 1.
- Beginning with 1.2 SP3, IBM Wave supports FTP Secure (FTPS). The FTPS protocol is described in RFC 4217 "Securing FTP with TLS". FTPS is used by default, unless IBM Wave detects that your installation is using standard FTP.
- SSH Key-based authentication for Linux. For information, see the following topics:
  - "Linux Login Security Options" on page 140
  - "NFS parameters" on page 122
  - "Security parameters" on page 125
  - "Changing User Preferences" on page 132
- A new menu option for "IBM Wave Database Actions" on page 99 is added to **Administrative**. "IBM Wave Database Actions" on page 99 includes the following options:
  - "Backup IBM Wave Database" on page 100.
  - "Regenerate IBM Wave Database Password" on page 101.
  - "Regenerate Encryption Keys" on page 101.
- The "Functionality parameters" on page 119 include new parameters that you can use to manage SMAPI authorization.
- The "Enterprise Directory parameters" on page 127 contains a new check box to "Allow user login without Group Allocation."
- "Changing User Preferences" on page 132 contains new information about controlling the SSH options and dismissing the "Submit Work Unit" message.

- "Security and IBM Wave user management" are divided into Chapter 6, “Security,” on page 137 and Chapter 7, “User management,” on page 147.
- Chapter 7, “User management,” on page 147 contains new topics about:
  - “Understanding user types and roles” on page 147.
  - “Overview of scopes and permissions” on page 148.
- Users who have the activate and deactivate permissions also have the relocate permission by default. For more information, see “Overview of scopes and permissions” on page 148
- A new exit, WaveConnectableGuestsExit, is added to test if guests' IP addresses are connectable. For more information, see “WaveConnectableGuestsExit - Connectable guests exit” on page 249.
- New audit messages are added to “HWVA0001E” on page 203 for signal activation, database password changes, cross-system cloning, and filtering.

Updated information

- Changes to the Background Task Scheduler (BTS):
  - In “The Background Task Scheduler (BTS) server” on page 9, the formula is updated and concurrent IBM Wave users is clarified to mean both GUI and CLI users.
  - In the “BTS Manager” on page 106, the **BTS Workers** table is now named the **Active BTS Worker Threads**.
  - In “BTS parameters” on page 117, the default number of BTS user worker threads is 2.
- To accurately reflect the proper AUTOLOG configuration and update the LANPROF WAVEPARM file, the following topics are updated:
  - “Making VSwitches permanent” on page 29.
  - “Configuring AUTOLOG” on page 35.
  - “Review the parameter files” on page 30.
- The information about “Functionality and Activation Levels and Activation Done signaling” on page 21 is updated with clarified examples and screens.
- “Configuring AUTOLOG” on page 35 is updated to match the current behavior of IBM Wave for z/VM.
- The "Security" topic, previously in Chapter 1, “Introducing IBM Wave for z/VM,” on page 1, is reorganized and moved to Chapter 6, “Security,” on page 137.
- The **Regenerate Encryption Keys** option moved to **Administrative** > **IBM Wave Database Actions**. For more information, see “IBM Wave Database Actions” on page 99.
- “Functionality parameters” on page 119 includes an update image with more details about functionality and activation levels.
- “IBM Wave user authentication” on page 143 was updated for currency and moved to Chapter 6, “Security,” on page 137.
- “Restoring the IBM Wave database” on page 100 is an updated task with changed information.
- Using the **Audit Log Display**, you can optionally filter the API and SSH Events, Messages from Automated Internal Scheduler Actions, or both. For more information, see “Displaying audit log events” on page 159.
- When Appendix H, “Customizing VM: Secure to use SMAPI,” on page 183, the permissions for **MAINT USER RULES** are MR.
- The exits are restructured. See Appendix O, “IBM Wave user exits,” on page 249.
- Appendix B, “A sample .csv file for importing guest attributes,” on page 167 is added.

Fix Pack 2 changes (December 2015)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 2.

New information

- IBM Wave supports the MariaDB on SUSE Linux Enterprise Server 12 (SLES12) and Red Hat Enterprise Linux 7 (RHEL7). For more Linux information, see the following topics:
  - "The Wave Linux server (WAVESRV)" on page 8
  - "Installation prerequisites" on page 51.
  - "Installing IBM Wave for z/VM" on page 62.
  - "Install the Wave Linux server (WAVESRV)" on page 62
- Before you add an external entity, such as a z/VM system to a CPC, see "Creating a new external entity" on page 85.
- Auditing is a new feature for IBM Wave for z/VM.
  - To control auditable events including the logging options, see "Audit Log parameters" on page 129.
  - For an overview about how auditable event logging works, see Chapter 8, "Audit Log Reporting feature," on page 159.
  - For the Audit Log message format and the IBM Wave messages, see "IBM Wave message format" on page 203 and Appendix N, "IBM Wave messages," on page 203.
  - All images for the IBM Wave parameters are new to reflect the addition of the Audit Tab and the movement of selected function. For more information, see Chapter 5, "System customization," on page 113.
- You can now activate periodic database backups. For more information, see "Backup IBM Wave Database" on page 100.
- You must run the "Init User for IBM Wave Use" action for all Linux virtual servers that are managed by IBM Wave. See "Initializing z/VM guests to work with IBM Wave" on page 25.

Updated information

- The topic "Freeze Changes" is changed to "Stop Updates", which matches the IBM Wave interface. For more information, see "Stop Updates" on page 16.
- The topic "z/VM guest and virtual server management" on page 20 is updated.
- Information is updated and added for "Deactivating Linux guests" on page 21.
- For Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), the installation guest require an access port type connection. See "Bare-metal installation (BMI) support" on page 25.
- The Virtual Network Segment (VNS) menu contains a "BTS Enabled" check box. For information, see "Virtual network segment" on page 30.
- To accurately reflect the **Attention Required** behavior, the topics about "The Attention Required mechanism" on page 47 and "Ignoring Attention Required entries" on page 48 are updated.
- "Linux Login Security Options" on page 140 is updated with guidance for the current Linux security options for IBM Wave.
- The Password Reset tool is renamed: "The password resetter utility" on page 142.
- Updated "Installation prerequisites" on page 51 and "Prerequisites for workstations that run IBM Wave" on page 53 to include:
  - New prerequisite z/VM APAR VM65744.
  - New requirement that 64-bit Java be used for the 64-bit version of Windows 7.
- Information about the "External Entities Manager" on page 85 is updated to accurately reflect that an entity is necessary to add a z/VM system, controller, or router to IBM Wave management.
- The "Dummy Region" must not be allocated larger than one cylinder. See "Running the Auto-Detect Wizard" on page 71 and "DASD storage status" on page 27.
- The "Audit Log Preview" field replaces the GUI threshold in the "GUI parameters" on page 116 tab.

- The "Log Dump Interval" and "SYSLOG" elements, previously displayed in the BTS tab for **IBM Wave Parameters**, now appear in"Audit Log parameters" on page 129.
- For the log options to work correctly, you must install the standard software on the BTS Linux Server. See "Wave server log options" on page 134.

Deleted information

- Information was removed from "Linux Login Security Options" on page 140 to reflect the current behavior and options for IBM Wave.
- The appendix topic about the "Initial Linux install" contained information that is no longer valid and was removed.

Fix Pack 1 changes (September 2015)

This edition includes changes to support product changes provided for the general availability of Wave 1.2 Fix Pack 1.

New information

- In "Linux Login Security Options" on page 140, there is new guidance for
  - Security products that must use alternate syntax.
  - Users who must have a home directory.
- For IBM Wave to Auto-Detect a z/VM system, you must define at least one DIRMAINT DASD group. See "Authorize DirMaint" on page 55.
- "Configure SMAPI" on page 54 contains important new guidance for specifying the Authorized API User as part of Auto-Detect process
- The following topics contain addition guidance to ensure American English (AMENG) is the only language you use when setting up the IBM Wave service machines.
  - "IBM mainframe requirements" on page 1
  - "IBM Wave service machines" on page 4
  - "Configuring IBM Wave service machines " on page 57
  - Appendix E, "Shared directory considerations for service machines," on page 173
  - Appendix F, "Considerations for the service machines when working with SSI," on page 177
- "Changing User Preferences" on page 132 contains new information about the PuTTY parameter and a new image.
- The "Installing IBM Wave for z/VM" on page 62 contains new guidance about "Review the parameter files" on page 30.
- "Wave server log options" on page 134 is a new topic about how to control the logging configuration.
- SUSE Linux Enterprise Server 12 (SLES12) and Red Hat Enterprise Linux 7 (RHEL7) are fully supported as managed guests. For more information, see the following topics:
  - "Initializing z/VM guests to work with IBM Wave" on page 25
  - Appendix A, "Linux distribution support," on page 165.

Updated information

- To handle the possibility of the unique ID for a directory changing, the following topics are updated:
  - "Unique directory identification" on page 12
  - "Relationship between z/VM systems and directories" on page 12
  - "z/VM Directory Manager" on page 88
  - "z/VM directory unique ID changes" on page 89

  - “Changing the source directory” on page 89
- The **Default NIC Address** is an attribute of the VSwitch. The definition is updated in “Virtual network segment” on page 30.
- The information in Chapter 2, “Installing and customizing IBM Wave,” on page 51 is reorganized and the following topics contain additional guidance:
  - “Installation prerequisites” on page 51.
  - “Configuring TCP/IP, SMAPI, and DirMaint” on page 53.
  - An optional step about “Setting up Performance Toolkit for z/VM” on page 56.
  - “Configuring IBM Wave service machines ” on page 57 and Appendix G, “Configuring VM: Secure,” on page 181.
  - “Installing IBM Wave for z/VM” on page 62, changes "Phase 1" and "Phase 2" to task-oriented topics and adds guidance for “Review the parameter files” on page 30 and what to do after you “Start IBM Wave for z/VM” on page 66.
- The topic about "Permanent and persistent VSwitch processing" was changed to “Making VSwitches permanent” on page 29 and the information is updated.
- For SUSE Linux Enterprise Server 11 (SLES11), mysql-MAX(5.0.67) is needed only for SLES11 SP2 and earlier releases. For information, see: “Install the Wave Linux server (WAVESRV)” on page 62.
- In the “BTS Manager” on page 106, the "Request Parms" column in the General Information tab is removed from the following tables:
  - "User Worker Stats"
  - "Internal Worker Stats".

Chapter 1. Introducing IBM Wave for z/VM

IBM Wave for z/VM is a provisioning and productivity management solution for managing virtual servers with z/VM.

The IBM Wave interface provides an innovative approach to the task of managing one or multiple IBM Z® systems. Each system can be configured with one or many z/VM instances that can each run virtual Linux servers.

IBM Wave has a unique graphical display of the virtual server environment and physical infrastructure that includes the following features:

- Physical servers (mainframes)
- z/VM instances (LPARs)
- Virtual Linux server objects
- Virtual Networks (Guest LANs and VSwitches)
- Virtual Servers to Virtual Network connections
- Storage volumes and storage groups

The graphical user interface (GUI) provides all the procedures and functions that are necessary for routine management and provisioning tasks and special operations. IBM Wave is intended to help eliminate the learning curve that is typically needed to manage and control z/VM and Linux guests.

IBM Wave abstracts the z/Architecture® and z/VM virtualization infrastructure to help Linux system administrators continue to manage their servers with the skill-set they currently possess. The convenience allows for day-to-day operations, along with large scale (virtual) hardware configuration changes, to be completed without the expertise that is often required from the z/VM system group.

IBM Wave provides the ability for IT organizations and service providers to simplify and automate z/VM administration, which makes it an ideal solution for medium-to-large scale consolidation projects in the IBM z/VM environment.

With IBM Wave for z/VM, the following are just a few of the tasks that you can click to complete:

- Perform basic z/VM guest actions, such as activate, deactivate, recycle, pause, and resume.
- Provision virtual resources, such as z/VM guests, network, and storage.
- Capture and clone virtual servers across every LPAR.
- Create and configure VSwitch and guest LANs.
- Connect virtual servers to virtual networks.
- Install Linux on a virtual guest.
- Relocate virtual guests with live guest relocation.
- Display and monitor page and spool and add and remove disks.
- Provision and track storage or free OSA and HiperSockets devices that use device pools.
- Manage storage at the z/VM level, such as dedicating devices, adding minidisks, and managing FCP storage.
- Manage Linux environment with the creation and expansion of LVM volume groups, regular partitions, and logical volumes.

IBM mainframe requirements

IBM Wave for z/VM 1.2 operates on and manages z/VM instances deployed on the following IBM Z family of hardware products:

- IBM z15™ Model T02

- IBM LinuxONE Model LT2
- IBM z15
- IBM LinuxONE III
- IBM z14 Model ZR1
- IBM LinuxONE Rockhopper II
- IBM z14
- IBM LinuxONE Emperor II
- IBM z13s
- IBM LinuxONE Rockhopper
- IBM LinuxONE Emperor
- IBM z13®
- IBM zEnterprise® EC12 (zEC12)
- IBM zEnterprise BC12 (zBC12)
- IBM zEnterprise 196 (z196)
- IBM zEnterprise 114 (z114)
- IBM z10 Enterprise Class (z10 EC)
- IBM z10 Business Class (z10 BC)

For information about the prerequisites for z/VM, Linux, IBM Wave, and the workstations, see "Installation prerequisites" on page 51.

For information about the steps to configure IBM Wave for z/VM, see Chapter 2, "Installing and customizing IBM Wave," on page 51.

Application architecture

As shown in Figure 1 on page 3, IBM Wave for z/VM implements a three-tier architecture that is made up of the following components:

Tier 1 - The IBM Wave graphical user interface (GUI) client
Controls the execution, attributes, and behavior of the Linux virtual servers and the managed z/VM resources as defined by the scopes and permissions of each IBM Wave user.

Tier 2 - The WAVESRV Linux guest server
The Linux server that runs the IBM Wave database, IBM Wave Application Server, and the IBM Wave Background Task Scheduler (BTS).

Tier 3 - The target virtualization platform (TVP) application programming interface (API)
The mediation layer that provides the interface into the managed environment.

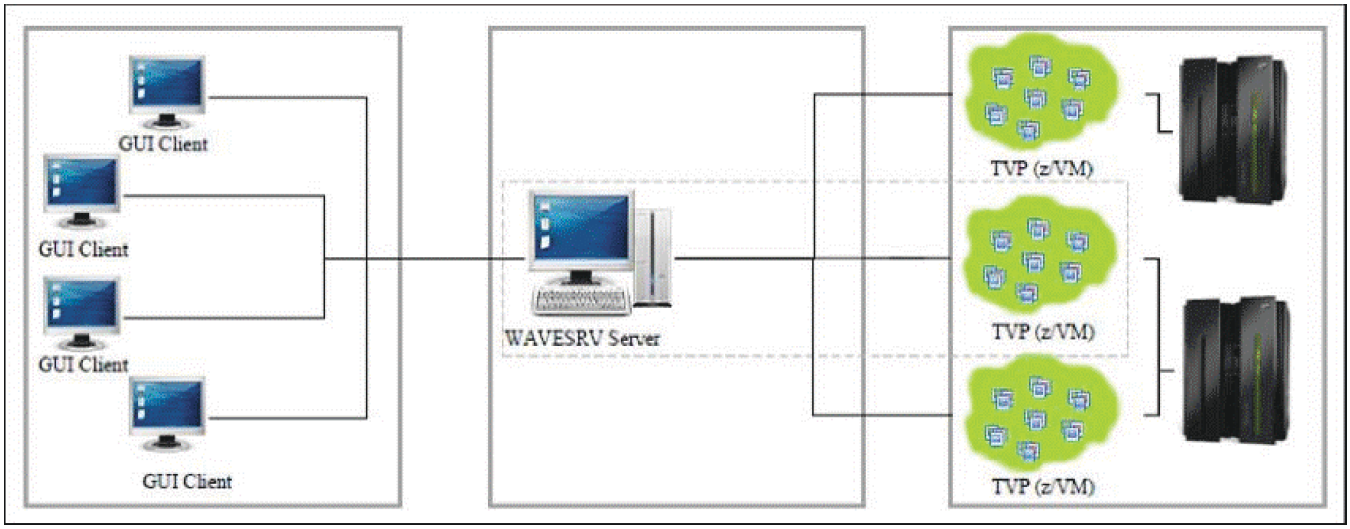


Figure 1. IBM Wave for z/VM's three-tier architecture

The interactions among the component tiers facilitate all of the features that comprise IBM Wave for z/VM.

Figure 2 on page 3 shows a high-level overview of how the tiers communicate with each other:

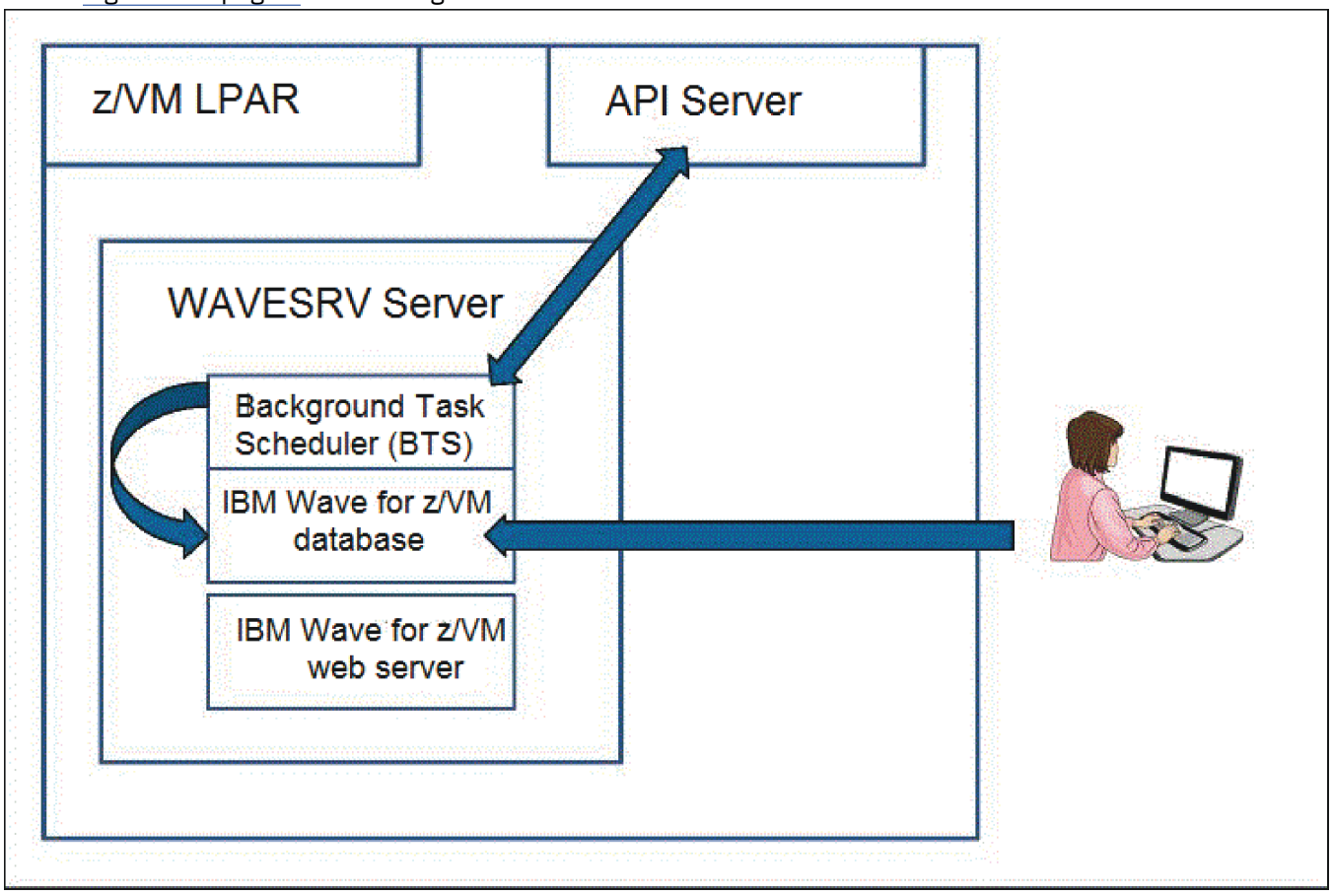


Figure 2. Communication among IBM Wave for z/VM's tiers

Supported target virtualization platforms (TVPs)

Any z/VM release that is supported by IBM is supported as an IBM Wave target virtualization platform (TVP). IBM Wave uses the z/VM Systems Management Application Programming Interface (SMAPI) to mediate requests from the Background Task Scheduler (BTS). Apart from SMAPI, IBM Wave also uses

service machines to take various actions that are not supported by SMAPI. The service machines are installed automatically during the auto-detect process when you add a z/VM system to IBM Wave management. IBM Wave supports DirMaint or an equivalent product as the directory manager for the z/VM system.

Note: Managing a z/VM system without a directory manager is not supported.

Interaction with the TVP

IBM Wave runs various queries and commands on the target virtualization platform (TVP) by using the IBM Wave service machines and the TVP management API. IBM Wave's interaction with the service machines uses the Background Task Scheduler (BTS), and can optionally be encrypted through SSL/TLS. The use of the TVP API requires authentication that uses a local user and password, which is referred to in this information as the *Authorized API User*.

For example, when z/VM is the TVP, a local z/VM Guest and its password are used. The Authorized API User is used to configure the IBM Wave prerequisites. When you add a z/VM system to IBM Wave management, IBM Wave requires you to input the Authorized API User and password for the TVP-API interaction.

IBM Wave service machines

This following information provides an overview of the IBM Wave service machines.

As part of the **Auto-Detect** process, when you add a z/VM System to IBM Wave management, three service machines are created and started on the z/VM System:

- WAVEWRKS - The service machine that runs REXX scripts, and runs **CP** and **CMS** commands that are necessary for some functions.
- WAVEWRKL - The service machine that runs some directory manager commands to facilitate some of the IBM Wave function.
- WAVEWRKC - The service machine that is responsible for the Cross System Clone (CSC) process. The service machine is either the sender or the receiver in the minidisk-streaming process. For more information about the minidisk-streaming process, see "Cross-system cloning and minidisk-streaming process" on page 34.

Important: The service machine requires American English, which must be set as `OPTION -LANG-AMENG`.

The service machines must be kept up and running in any z/VM LPAR that is managed by IBM Wave. IBM Wave monitors the service machines and alerts you if any errors occur. Note when a CSC process is in progress, the WAVEWRKC service machine is not monitored, and a warning message is sent to IBM Wave users.

Note: WAVEWRKS, WAVEWRKL, and WAVEWRKC are the default names for the service machines. In IBM Wave, you can configure the service machines with different names.

IBM Wave uses the z/VM Systems Management Programming Interface (SMAPI) in addition to the service machines.

The IBM Wave user interface

The IBM Wave user interface client is the first tier of the IBM Wave three-tier architecture.

The IBM Wave user interface is based on Java and runs as a *Java Web Start* application. The WAVESRV server has a minimal web server that allows the download of the client. Chapter 2, "Installing and customizing IBM Wave," on page 51 contains a section about how to transfer the GUI client deployment to another web server, if needed.

The client is the user interface for managing your z/VM environment. The client interfaces with the Background Task Scheduler (BTS) on the WAVESRV server to make updates to the database and managed resources. Currently, only one z/VM LPAR can be viewed and interacted with at one time, but you can define unlimited z/VM LPARs in IBM Wave. Some actions for objects are updated only in the IBM

Wave database. Other actions might require updates to the z/VM LPAR itself, or to virtual guests that are running on the z/VM LPAR.

Actions that involve updates or queries to the z/VM LPAR are done by using the standard z/VM System Management API (SMAPI), or by using the IBM Wave service machines.

Figure 3 on page 5 is an overview of the elements in the IBM Wave user interface.

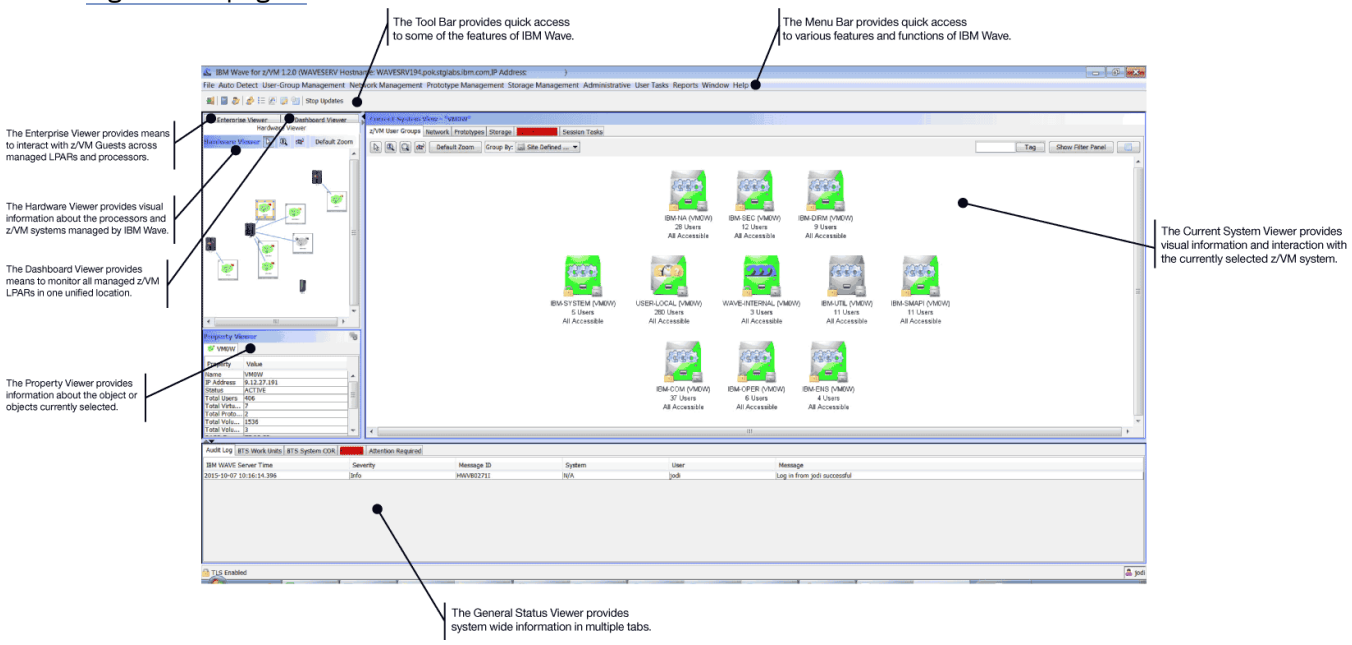


Figure 3. The IBM Wave user interface

Overview of the IBM Wave client

The following topic is an overview of some of the technology that the IBM Wave client uses.

IBM Wave for z/VM is a Java Web Start application. The web server on the WAVE server contains a link that downloads and installs the IBM Wave graphical user interface (GUI) client on the workstation. The process also creates a link on the desktop and in the Windows Start menu. After you install the application, double-click the icon or menu shortcut to start IBM Wave for z/VM.

- The Java Web Start (JavaWS) framework includes an automatic search for updates. After the maintenance [1] is applied to IBM Wave, the IBM Wave client is updated. A reinstall is not necessary. The next time that IBM Wave starts, it is updated automatically.
- For the procedure to initially start IBM Wave, see “Start IBM Wave for z/VM” on page 66.
- If you need to change the IP address for the server, see Appendix D, “Changing the IBM Wave server IP address or host name,” on page 171.

Single glance technology and the GUI engine

The GUI engine is responsible for facilitating all interaction and viewing of the z/VM and IBM Wave objects. IBM Wave contains technology to help make it easier to manage your z/VM complex thanks to the complex layout algorithms, which can display even large z/VM complexes in an instant. In large environments, zooming and rotating diagrams can help make the layout easier to view.

IBM Wave's Single Glance Technology provides an informational view of the objects on the screen. Every icon that represents a z/VM or IBM Wave object contains status information. In general, all of the icons are informational. For example, an icon for a z/VM guest indicates:

- Type of the user (Linux virtual server, CMS user, service machine, an IBM Wave virtual server).
- Current status (Active, Inactive, Starting, Shutting down, Cloning, and more).

[1] To apply maintenance, use the latest fix pack from the IBM Support Portal.

- Connectivity or "connectable" (IBM Wave detects when the virtual server has at least one IP interface that is reachable from the WAVESRV server).

The IBM Wave GUI engine is in constant contact with the WAVESRV server to ensure that your GUI display represents the current state of your environment. The GUI engine also facilitates all of the interaction with the z/VM LPAR. For example, to connect a virtual server to a virtual network, all the Linux system programmer must do is select the appropriate tool to create a connection between the virtual server and the virtual network.

For description of the icons, see:

https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_the_main_viewer.html

Session tasks

IBM Wave enables the user to run a set of actions on multiple virtual servers in parallel. The multitasking capability depends on the system configuration for the:

- Version of the SMAPI server in the z/VM System
- Number of Directory Manager service machines
- Number of SMAPI worker servers
- Multitasking capability of the workstation.

Running >multiple actions can be a lengthy process, so IBM Wave allows the action windows to be hidden and displayed. The **Session Tasks** viewer contains a list of the multiple-task windows that are generated for the session and indicates the status and progress. Double-click the corresponding entry to view the display or remove a multiple task action from the view.

For more information, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_the_session_tasks_viewer.html.

Important: Session tasks must in no way be considered background, server-side tasks. Although the implementation suggests that these processes are running in the background, they are running on the workstation, and only the windows are minimized. Exiting IBM Wave forcibly ends the processes. IBM Wave warns the user when they exit the GUI that there are active multiple task actions in progress.

Personalization capabilities

IBM Wave contains a personalization mechanism that saves information about the user preferences in the IBM Wave database.

Currently, the personalization mechanism saves the following data:

- **Last screen configuration** - When you log off, the personalization system saves the screen configuration. When you log in again, the screen reverts to the saved configuration.
- **Last selected hardware element** - When you log off, the currently selected hardware element (CPC or z/VM System) is saved. The next time you log in, that hardware element is automatically selected. If the selected hardware element is a z/VM System, then the **Current System Viewer** is populated with the information about that system.
- **Last Selected View within the selected hardware element** - If the selected hardware element is a z/VM System, then the current view within that currently selected z/VM System is saved. During the next login, the application switches to that view.
- **Last Selected Filters** - The filters that are applied anywhere in the application are saved and are implemented during the next login.
- **Selected columns in the Table Viewers** - You can customize the table viewers so that certain columns are hidden. Hidden columns are saved across sessions.

  Note: The order of columns is not saved across sessions.

- **IBM Wave User Preferences** - You can customize the IBM Wave experience by changing some basic IBM Wave User preferences such as the use of layout animation, the BTS log size, and more. For complete information, see “Changing User Preferences” on page 132.

Single User Mode

The **Single User Mode** is useful when an administrator must apply maintenance to IBM Wave.

An IBM Wave Administrator can enter the **Single User Mode** in which no other user can log in to the system.

To toggle the **Single User Mode**, click **Administrative > Toggle Single User Mode**.

IBM Wave internal messaging mechanism

IBM Wave provides an internal messaging message mechanism for IBM Wave administrators to send internal messages to all users who are logged in. For example, the messaging mechanism is useful when an administrator wants to shut down IBM Wave for maintenance.

The messages are transferred by using IBM Wave resource serialization (WRS) technology.

For more information, see “Broadcast Message to IBM Wave Users” on page 110 and “Send Message” on page 109.

Locking and unlocking an entity or object

With the proper permissions, you can lock or unlock an entity or object to prevent changes from being made. The locking mechanism ensures that no changes are made to the locked entity or object.

Note: The lock and unlock mechanisms are metadata attached to the entity or object, which means that the lock prevents changes that are made in IBM Wave only. For example, a z/VM administrator can change a locked z/VM Guest by entering commands directly to z/VM. After a z/VM administrator changes a locked entity, it is reflected in IBM Wave the next time an entity update is run by the Background Task Scheduler (BTS).

IAN and CAAP technology

IBM Wave provides you with the ability to attach an intelligent active note (IAN) to managed objects. An IAN is similar to a note that is attached to a hardware element in a server room. You can attach an IAN to any managed object (such as z/VM Guests, Virtual Networks, direct access storage devices (DASD) components, IBM Wave Scripts, IBM Wave Users, and others). An IAN displays in a tooltip when you hover over the object.

The IBM Wave ***Context Aware Action-Prevention (CAAP)*** technology integrates with an IAN to protect against unwanted actions or behavior. When a managed object has an attached IAN and the contents refer to an action, a special warning pane displays if a user acts on that managed object. (Even when the IAN does not refer to an action, IBM Wave displays the pane.)

Figure 4. IAN with deactivate that uses CAAP in bold

An IAN that refers to an action is highlighted in bold, and the word that CAAP identifies as referring to the action is also highlighted in bold.

In Figure 4 on page 8, an IAN with the text "`Please do not` **`deactivate`** `this server`" is attached to a z/VM Guest. When a user tries to use **deactivate** on the server, the pane displays and the tab that contains the IAN is highlighted in bold.

You can update or delete and IAN from all managed objects. You cannot change the existing contents on an IAN, but you can append text to the end. When an IAN is updated, a title with the IBM Wave user name and a date and time stamp is automatically added to the IAN.

The Wave Linux server (WAVESRV)

The WAVESRV server is the second tier in IBM Wave's three-tier architecture. The WAVESRV server runs Linux. The server runs as a virtual server within a z/VM LPAR, or as a Linux partition on PR/SM.

The following two options are available for WAVESRV implementation:

- **Single server** - One WAVESRV server is installed for the entire complex. The single-server configuration is preferred because it reduces IBM Wave maintenance and processor usage. The single-server option requires that the WAVESRV server has IP access to all managed CPCs, z/VM LPARs, and IBM Wave GUI actions.
- **Multiple servers** - Multiple WAVESRV servers are installed in the complex. The multiple server option is for customers who, for internal structure or policy, cannot conform to the single-server option requirements. The IBM Wave GUI client can connect to only one WAVESRV server at a time.

No matter the option you select, the installation is the same.

Notes:

1. The WAVESRV server requires American English. No other languages are supported.
2. You can have numerous WAVESRV servers that are managing different z/VM LPARs.
3. IBM Wave does not support two WAVESRV servers that are managing the same LPAR. Currently, no interaction happens between WAVESRV servers. In theory, it is possible to have two different WAVESRV servers, both of which manage the same z/VM LPAR. However, concurrent changes can be

made by different users to the same entities in that z/VM LPAR, which can lead to unexpected behavior.

The WAVESRV server has several roles:

Database server
The IBM Wave main database is stored on the WAVESRV server. The database contains information about all z/VM elements that are managed by IBM Wave, the IBM Wave regular log, the IBM Wave users, and more.

Background Task Schedule (BTS) server
All background tasks are run within the WAVESRV server. For more information, see "The Background Task Scheduler (BTS) server" on page 9.

Web server
The WAVESRV server runs a minimal Apache web server to allow the installation of IBM Wave GUI client. For more information, see "The IBM Wave user interface" on page 4.

Application server
The WAVESRV server runs an embedded WebSphere Liberty server. The WebSphere Liberty server hosts the IBM Wave application server that enables the IBM Wave RESTful APIs. For more information, see Chapter 3, "Wave APIs and WebSphere Liberty," on page 83.

For instructions about installing and customizing the WAVESRV server, see Chapter 2, "Installing and customizing IBM Wave," on page 51.

The Background Task Scheduler (BTS) server

The IBM Wave Background Task Scheduler (BTS) constantly updates the IBM Wave database with live data from all the managed and active z/VM LPARs. The GUI client uses the BTS to get information from the IBM Wave database and the z/VM API Server. Even when changes are made outside of IBM Wave, by a z/VM administrator, the changes are reflected in the IBM Wave client.

The IBM Wave BTS is built as a service executor that receives requests through a proprietary communication protocol (based on TCP/IP), and handles the requests by using worker threads. To customize the port on which the BTS listens for requests, a Wave server Linux administrator uses the procedure in step "8" on page 64 (*during installation*) or the procedure in "Upgrading IBM Wave for z/VM" on page 68 (*when updating Wave*).

Communication between the BTS and the GUI stations are encrypted with the SSL/TLS protocols.

The BTS dispatches work on BTS Worker Threads. IBM Wave contains two types of BTS worker threads:

1. **User Worker Threads** - BTS user worker threads are responsible for running user-generated BTS work units and requests. The number of user worker threads is customizable. To determine the number of user worker threads, use the following formula:

   ```
   <Number of concurrent IBM Wave GUI or CLI Users> * 2
   + <Number of concurrent clone operations>
   * <Maximum number of minidisks in the source guest for cloning>
   ```

   For example, in an environment with three concurrent IBM Wave users logged in (users of the GUI or the CLI) with two clone operations that are running in parallel, and each one is cloning a guest with four minidisks, the number of user worker threads is `(3*2) + (2*4) = 14`.
2. **Internal Worker Threads** - BTS Workers are responsible for running internal IBM Wave work units and requests, such as the periodic tasks. The number of internal worker threads cannot be customized. The number of Internal BTS worker threads depends on the number of z/VM Systems managed by the BTS (among other aspects) and is represented by the following formula:

   ```
   2 + <Number of managed z/VM Systems> * 4
   ```

   For example, in an environment with three managed z/VM Systems, the number of internal BTS worker threads is `(3*4) + 2 = 14`.

In general, the number of User Worker Threads allows the system administrator to control how much work can be moved onto the system by IBM Wave users. If you want to lower the impact of IBM Wave users on the system (in terms of performance), you can define a lower number of worker threads. Doing so means that users might have to wait for their requests to move up the queue and be dispatched on a free BTS worker. If you define a higher number of worker threads, more work can be pushed in parallel. Users have less or no wait time, but the performance impact to the system might be higher.

You can change the number of User Worker Threads can by modifying the IBM Wave Parameters. For more information about the BTS parameters, see Chapter 5, “System customization,” on page 113.

If a higher number of workers is specified, worker threads are created dynamically. If a lower number is specified, worker threads are shut down gracefully, which means that you can change the number of workers upon request. For example, if you usually do not clone at all, and have two IBM Wave users who are working in parallel, the defined number of workers is four. However, if suddenly you must initiate a clone request, you can change the parameter to a higher number. Then, after the clone operation finishes, you can reduce the number.

You can monitor the BTS activity by using the **Administrative** > **BTS manager** option from the **IBM Wave Main Menu**. For more information, see “BTS Manager” on page 106.

The BTS has an internal scheduling mechanism that schedules requests at user-defined intervals.

Common output repository

The IBM Wave common output repository (COR) is a central location that contains all of the output generated by the Background Task Scheduler (BTS), and all of the BTS requests. The COR is consists of COR entries. Currently, one COR entry that is named the Log COR entry is used to log all the activity for a specific BTS Request. The second special type of the Log COR entry is the System COR entry, which logs all of the activity done by the BTS. You can view the system COR entries in the **System COR Entry Viewer** tab from the **Current System Viewer**.

Other Log COR entries can be viewed from the **BTS work units details** window, which is accessed by double-clicking a specific BTS work unit, and then selecting the specific BTS request. COR Entries can be manually deleted from the BTS work unit details view. They are automatically deleted when the BTS work unit that owns the BTS request to which the COR Entry belongs is deleted from IBM Wave. Each BTS work unit is kept in the IBM Wave database for an amount of time that is specified by the IBM Wave Administrator. At the specified time, the BTS work unit and its BTS requests, and any associated COR entries are deleted from the IBM Wave database.

BTS work unit requests

Every Background Task Scheduler (BTS) request is run under a BTS work unit. You can view the BTS work unit in the **General Status Viewer** > **BTS Work Unit** tab. A BTS work unit can have one or more BTS requests. Each BTS request is allocated at least one COR entry (a log COR entry) when it is started. The COR entry logs all the activity of the BTS request.

To view the Log COR Entry for a specific request, double-click the work unit name in the **BTS Work Unit** tab and then select the specific BTS request.

BTS work unit scheduling

The IBM Wave Background Task Scheduler (BTS) contains a scheduling mechanism that automatically schedules certain requests to run at specific intervals. The scheduled requests provide the following functions:

1. Monitor internal IBM Wave components. Some of the scheduled requests monitor the status of the IBM Wave service machines in each managed z/VM system and the status of the file system in the WAVESRV server, clean BTS work units that expired, and more.
2. Update the IBM Wave database with information from the managed z/VM systems on a periodic basis (z/VM guests, virtual networks, storage status, and more).

Some of the requests are global and are added only one time. Other requests are CPC or z/VM LPAR-specific and are automatically added to the scheduler when a CPC or z/VM LPAR is added to IBM Wave management. To control the scheduling of entries, use the **Administrative** > **BTS manager** option.

To change such parameters as the sleep interval or debug level, go to **Administrative** > **Manage Parameters** and click on the **BTS** tab.

You can view the BTS output in the **General Status Viewer** > **BTS Log**.

To change the level of messages that are displayed in the BTS **Message Level** pane, use the **User Tasks** > **Change IBM Wave User Preferences**.

To view the output of specific scheduled requests, open the **BTS System COR** from the **General Status Viewer** > **BTS Work Unit**.

BTS task tracking and failure notification

The Background Task Scheduler (BTS) retains statistics for all periodic internal tasks, which include the following events:

- Number of times a task runs
- Number of errors a task encounters
- The last time the task ran
- The last time the task encountered an error
- The last time the task statistics were reset.

If a periodic task fails, the BTS sends a notification to all open GUI clients. The notification causes the **BTS Log** tab, in the **General Status Viewer**, to be colored red. After the statistics are reset or you ignore the task, the BTS sends another notification that clears the flag and resets the tab color to normal.

The statistics for the periodic tasks can be viewed and manipulated by using **Administrative** > **BTS Manager** window. For more information, see “BTS Manager” on page 106.

BTS directory manager work unit sampler

The Directory Manager work unit sampler is a component within the BTS that is responsible for periodically querying the status of active Directory Manager work units and updating the status. For more information about Directory Manager work units, see “Directory manager generated work units” on page 36. You can set the sampling interval through the IBM Wave Parameters.

Note: The Directory Manager work unit sampler actively samples work units only when the work units are generated by BTS Requests. The sampler cannot track Directory Manager work units that were generated by actions taken outside of IBM Wave.

BTS Live Guest Relocation sampler

The Live Guest Relocation (LGR) sampler is a component within the Background Task Scheduler (BTS). The LGR sampler is responsible for periodically sampling the progress of active LGR BTS requests, and updating the status of the requests. To set the periodic sampling interval, go to **Administrative** > **Manage Parameters** and click on the **BTS** tab.

Note: The LGR sampler actively samples LGR requests only when the requests are generated by BTS Requests. LGR processes triggered outside of IBM Wave for z/VM are not tracked.

Other periodic BTS tasks (such as the **z/VM Guest Status** periodic task) identify when a guest moves from one system to another. As a result of an LGR request, an update for the z/VM guest state is made in the database.

Shared directory support

Shared directory support is provided with IBM Wave for z/VM.

IBM Wave supports sharing a z/VM Guest directory between two or more z/VM LPARs providing that the installed directory manager supports the configuration. IBM Wave can uniquely identify the existing z/VM Guest directories and automatically associate z/VM Systems that are added to IBM Wave management with their respective directories. Guests that are detected are marked with a "Logon Eligibility" flag that reflects the value that is specified in the prefix form of the SYSAFFIN statement, if one exists.

Note: Limited support exists for the internal form of the SYSAFFIN directory statement, as outlined in “SYSAFFIN statement support” on page 12.

Unique directory identification

During the **Auto-Detect** process, IBM Wave retrieves a unique identifier (unique ID) for the directory with which the z/VM System works. Auto-Detect occurs when you add a z/VM System to IBM Wave management and during any subsequent **Auto-Detect** processes that occur for an existing suspended system. The unique ID is based on the unique identification of the DASD volume on which the source directory is stored.

When the source directory is shared, it must be stored on a DASD volume that is shared and accessible across all of the z/VM Systems that are sharing the directory. The unique ID must be the same for all z/VM Systems that are sharing the directory.

Note: For installations that use DIRMAINT as the directory manager, the unique ID is based on the unique identification of the DASD volume on which the DIRMAINT cluster files are installed.

Relationship between z/VM systems and directories

For each unique directory that IBM Wave detects, a directory record is saved in the IBM Wave database. Any z/VM System that uses the directory is automatically associated with the directory record. IBM Wave monitors the directory contents, but not its location, on a periodic basis. Therefore, if the unique identification ("Unique ID") of the directory changes, the changes are not reflected in the database or the IBM Wave clients, until the **Auto-Detect** process is manually run on each system associated with the directory.

To view the system's directory, use the **Administrative** > **Site Management** > **z/VM Directory Manager**. You can also select a specific z/VM System to view its directory.

For more information, see “z/VM Directory Manager” on page 88 and “z/VM directory unique ID changes” on page 89.

SYSAFFIN statement support

IBM Wave fully supports the prefix form of the SYSAFFIN statement, which can be used to limit the logon eligibility or the existence of specific guests on z/VM Systems that are sharing the directory. Based on the existence of guests, the value that is specified in the prefix SYSAFFIN statement, or both, IBM Wave might keep one or more occurrences of the guest in the database (up to the number of z/VM Systems that are sharing the directory).

The following SYSAFFIN parameters are supported:

LOGON_AT
IBM Wave creates an occurrence for the guest for all z/VM Systems sharing the directory, but logon eligibility is limited to the systems provided as the parameter to the **LOGON_AT** keyword.

NOLOG_AT
IBM Wave creates an occurrence for the guest for all z/VM Systems sharing the directory. Logon eligibility is limited to the systems not provided as the parameter to the **NOLOG_AT** keyword.

EXISTS_AT
IBM Wave creates an occurrence for the guest for all z/VM Systems sharing the directory. However, guest occurrences that are created on systems that are not provided as a parameter to the **EXISTS_AT** keyword are not visible to the user.

No SYSAFFIN specified
IBM Wave treats such a case as if **SYSAFFIN EXISTS_AT *** was specified.

For each z/VM Guest detected by IBM Wave, two types of records are created:

1. **Directory Record** - The **Directory Record** contains data that is common to all occurrences of the guest. The data includes IBM Wave metadata, the Guest's directory entry, creation, and update time stamps, and other data. If any of the data is changed, it is automatically reflected in all occurrences of the guest. There is exactly one directory record for each guest in a directory, regardless of the number of z/VM Systems sharing the directory.
2. **Per-System Record** - The **Per-System Record** contains data that is or can be relevant to a specific occurrence of a guest in a specific z/VM System sharing the directory by use of the SYSAFFIN or SUBCONFIG statements. The data includes storage (memory) definitions, Disk space, CPU count, and others. The number of per-system records for each guest is exactly the number of z/VM Systems sharing the directory.

For example, in a complex that has three LPARs sharing a single directory, each guest in the directory has one directory record, and three per-system records.

The per-system records make IBM Wave aware of the different configurations that are given to z/VM guests when brought up in different z/VM Systems that are sharing the directory. Therefore, inventory-related functions, such as Disk Mapping, Guest Details View, Network Topology, and others, might show different information. It depends on which occurrence of the guest is selected for the action.

Any IBM Wave action or function that changes the directory of guests is not valid for guests that have the internal form of the SYSAFFIN statement in their directory definition. The actions are also not valid for IDENTITIES. The actions include connect and disconnect to and from Virtual Network Segments, Manage Storage, and other actions.

Guest logon eligibility

IBM Wave marks guests as eligible for logon based on the following criteria:

In a directory that is not SSI-ready or SSI-enabled:
The existence of **and** or **or** parameters that are provided to the prefix form of the SYSAFFIN statement, which can optionally be specified in the user directory.

For more information about how IBM Wave handles various options in the SYSAFFIN statement, see “SYSAFFIN statement support” on page 12.

In a directory that is SSI-ready or SSI-enabled:
USER entries are always marked as **EXISTS_AT *****, while the logon eligibility for IDENTITIES depends on the associated BUILD and SUBCONFIG statements.

Guests that are not eligible on a particular system are displayed with dimmed icons, and no action can be run against them. Use the **Hide Ineligible Guests** preference to control whether guests that are ineligible are displayed at all. By default, the **Hide Ineligible Guests** preference is turned on to avoid cluttering the display and to avoid unnecessary memory and CPU consumption.

Note: If the **Hide Ineligible Guests** preference is turned off, IBM Wave displays all guests per z/VM System regardless of the value of the SYSAFFIN statement except for the **EXISTS_AT** keyword.

With directories that are SSI-ready or SSI-enabled, the IBM Wave behavior for IDENTITIES and BUILD and SUBCONFIG statements is identical to the one described in Table 2 on page 14 for the SYSAFFIN **EXIST_AT** parameter statement. For example, in an SSI cluster with three LPARs (A, B, and C), when an IDENTITY contains only BUILD statements for two of the LPARs (A and B), the logon eligibility is the same as if it was a USER with SYSAFFIN **EXISTS_AT** A B.

Table 2 on page 14 describes the behavior of the application based on the value of the **Hide Ineligible Guests** preference. The configuration is based on two z/VM Systems (SYSA and SYSB) that share a directory, and applies to a specific guest.

Table 2. Guest eligibility based on directory contents and preference settings

Directory Contents and preference settings	Display when SYSA is selected in the Hardware Viewer	Display when SYSB is selected in the Hardware Viewer	Number of occurrences of the guest in the Enterprise Viewer
SYSAFFIN LOGON_AT SYSA, Preference off	Eligible	Ineligible	Two
SYSAFFIN LOGON_AT SYSA, Preference on	Eligible	Guest not displayed	One
SYSAFFIN LOGON_AT SYSB, Preference off	Ineligible	Eligible	Two
SYSAFFIN LOGON_AT SYSB, Preference on	Guest not displayed	Eligible	One
SYSAFFIN NOLOG_AT SYSA, Preference off	Ineligible	Eligible	Two
SYSAFFIN NOLOG_AT SYSA, Preference on	Guest not displayed	Eligible	One
SYSAFFIN NOLOG_AT SYSA, Preference on or off	Eligible	Guest not displayed	One
SYSAFFIN EXISTS_AT SYSB, Preference on or off	Guest not displayed	Eligible	One
No SYSAFFIN specified, Preference off	Eligible	Eligible	Two
No SYSAFFIN specified, Preference on	Eligible	Eligible	Two

Other elements shared across the directory

Some directory managers share more than z/VM Guests. Elements such as DASD Groups and Regions, Prototypes, and Profiles can also be shared. IBM Wave contains the following special handling of the elements:

- **Prototypes and Profiles** - Prototypes and Profiles are handled in the exact same manner as z/VM Guests.
- **DASD Groups and Regions (DIRMAINT)** - For each DASD Group in a shared directory complex, an eligibility flag determines which z/VM System is valid for the DASD allocations. The eligibility flag is computed by IBM Wave based on the DASD Regions that are assigned to the Group, and the DASD Volumes on which these DASD Regions are defined. (As illustrated in Figure 5 on page 14.)

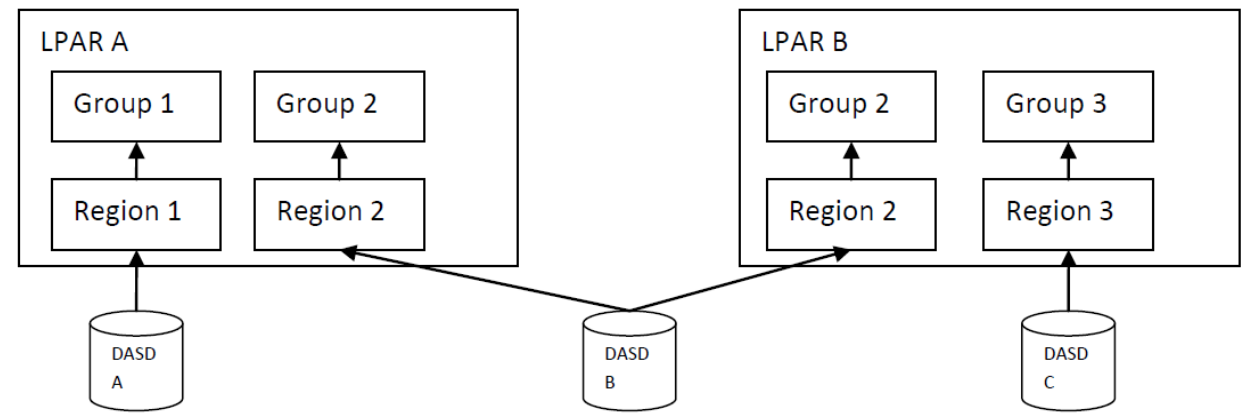


Figure 5. Private and shared DASD groups

- DASD A is private to LPAR A.
- DASD B is shared between both LPAR A and LPAR B.
- DASD C is private to LPAR B.

The output of various query commands to DIRMAINT produce the same output whether issued on either LPAR, IBM Wave identifies the relationship and marks

- Group 1 as eligible in LPAR A.
- Group 2 as eligible in both LPAR A and LPAR B.
- Group 3 as eligible in LPAR B.

Visualization of shared directory

The **Hardware Viewer** shows the z/VM Systems that share a directory grouped together. The group's name is the directory name, which is set by default. You can modify the name in the **Administrative** > **Site Management** > **z/VM Directory Manager**.

Wave resource serialization and shared directory

IBM Wave considers a guest to be singular in a directory, but replicated per z/VM System in which the guest exists. Theoretically, if a guest is eligible on more than one z/VM System it can be changed by two separate IBM Wave users simultaneously. To prevent such a situation from occurring, guest integrity is maintained by using IBM Wave resource serialization (WRS) technology. The serialization is done on a logical "directory" occurrence of the guest, so changes to shared guests are serialized. The same applies to any other element shared across the directory.

Single system image and live guest relocation support

IBM Wave fully supports z/VM systems which are members of Single System Image (SSI) clusters. Such systems are grouped together in the Hardware Viewer, in a similar fashion to z/VM systems sharing a directory. The color of the group in the Hardware Viewer (light blue) indicates that they are a part of an SSI cluster, and the name of the group will reflect the name of the SSI cluster as it appears in the output of the Q SSI CP command.

Live Guest Relocation (LGR) [2] is fully supported, under the same conditions, provisions and limitations imposed by z/VM. LGR can be run against one or more guests, as a multiple task action, either by selecting the appropriate action from the pop-up menu when right-clicking the guests or by dragging and dropping the guests in the Enterprise Viewer when the **Group By** setting is set to **z/VM System**.

The LGR process generates a BTS workunit that can be tracked from the BTS workunit viewer.

Limitation for Identities

IBM Wave provides limited support for managing Identities. All actions which trigger a change to the directory (Connect/Disconnect, Manage Storage, Certain aspects of the "Update" action, convert to Prototype, clone, duplicate configuration, etc.) are disabled for Identities. This is similar to the limitations imposed on z/VM Guests using the internal form of the SYSAFFIN statement as illustrated above.

Special consideration for spool/page DASD

When you use SSI clusters, ensure the spool/page DASD is formatted with the SSI cluster name.

IBM Wave provides function to add a pre-formatted spool/page devices to a z/VM System. If the z/VM System is part of an SSI cluster, the spool/page DASD must be properly formatted with the SSI cluster name.

[2] Live Guest Relocation (LGR) definition: In a z/VM SSI cluster, a running guest virtual machine can be moved from one member to another. This process is called live guest relocation. The functions for initiating and managing guest relocations are provided by the CP VMRELOCATE command.

Change synchronization and serialization

Because IBM Wave is a multi-user environment, it has special change handling to synchronize the changes made to objects that it manages. Internal object locking prevents several IBM Wave users from changing the same object at the same time.

- Change synchronization is handled by the "Automatic Change Synchronization" on page 16 (ACS) technology.
- Object locking is handling by the "Wave Resource Serialization" on page 16 (WRS) technology.

Automatic Change Synchronization

An overview about how ACS technology works with the Background Task Scheduler (BTS).

IBM Wave for z/VM uses Automatic Change Synchronization (ACS) technology to facilitate the propagation of changes in the system. All change propagation events start from the Background Task Scheduler (BTS) and are forwarded to the IBM Wave GUI clients.

For example, when an IBM Wave User creates a new VSwitch, the change is run within the BTS as a BTS Request. After the request finishes, the change is reported to all GUI clients connected to the BTS, based on the scope of the IBM Wave User who is running the client. The GUI client receives the change report, and propagates it to all relevant GUI viewers and windows to reflect the change. With the VSwitch example, the **Network Viewer** is refreshed to show the new VSwitch. IBM Wave Users that do not have scope for the added VSwitch do not receive or see the change.

Show Changes Log

The Automatic Change Synchronization Log (ACS Log) shows all of the changes that are reported to the GUI from the time of login. The log shows the change type (add, delete, and update) with a reference to the managed object that changed.

To view all of the changes log, that are reported to the GUI client, from the **IBM Wave Main Menu** click **User Tasks** > **Show Changes Log**.

Stop Updates

Stopping updates is useful when you want to rearrange items in a certain viewer in preparation for screen captures, printing, or to clarify your viewing preferences.

To stop updates in the IBM Wave client, click **Stop Updates**. The **Stop Updates** status changes to **Process Updates**.

When you click **Stop Updates**, all IBM Wave logging actions are disabled and all incoming change reports are held in the queue. You must click **Process Updates** to resume normal processing. When the session resumes, all change reports that accumulated in the queue are processed in FIFO order.

Wave Resource Serialization

Wave Resource Serialization (WRS) serializes access to z/VM objects to avoid concurrent update problems.

Wave Resource Serialization (WRS) is a mechanism that is designed to serialize access to IBM Wave and z/VM resources. Because IBM Wave is a multi-user environment, two different IBM Wave Users can perform contradicting actions on the same z/VM Object.

For example, User A decides to activate a server, and User B decides to delete the server. Another example is when a virtual Server is connected to a VSwitch. Because IBM Wave automatically suggests the IP addresses, the requests are managed by the WRS mechanism. If two IBM Wave Users try to connect two different virtual servers to the same VSwitch, they never receive the same suggested IP address.

IBM Wave administrators can view and interact with WRS elements by clicking **Administrative** > **View WRS Elements**.

Metadata objects and entities

IBM Wave provides several methods to add metadata objects and entities to be managed by IBM Wave. Some objects and entities require more associated metadata than others.

The following objects must first be defined as a managed entity before IBM Wave can manage them:

1. z/VM Systems that are managed (or not) by IBM Wave.
2. Other entities such as routers, storage controllers, and more.

Other entities can be associated with certain IBM Wave objects such as device pools and virtual network segments.

For more information about metadata objects and entities, see the following topics.

- “External Entities Manager” on page 85
- “Automatic Guest Classification” on page 37.

Project

You can assign z/VM Guests from different LPARs to the same project.

IBM Wave provides you with the ability to define projects, and then assign z/VM Guests or virtual servers to the project. Using the "Group-By" view, you can then view the guests that are grouped by "Project", or search and filter by project. Project definitions are metadata and do not interact with z/VM.

Projects can be defined across systems and LPARs, which means you can assign z/VM Guests from different z/VM LPARs to the same project.

For example, your site defined a project that is called **PROJECT A**, which has production virtual servers, development virtual servers, and QA virtual servers.

- The production virtual servers are in the "Production z/VM LPAR".
- The development virtual servers are in the "Development LPAR".
- The QA virtual servers are in the QA "z/VM LPAR".

When all of the LPARs are managed by IBM Wave, you can assign all of the virtual servers to **PROJECT A**.

An administrator can define, update, and delete a project by using the **Administrative** > **Project Manager**. For more information, see “Project Manager” on page 103.

Site Defined Groups

You can work with **Site Defined Groups** by using the **z/VM Guests tab** in the **Enterprise** viewer.

IBM Wave can help make complex z/VM environments easier to manage. One feature is the **Site Defined Groups**, which are groups of z/VM Guests. Every z/VM Guest belongs to a Site Defined Group (no guests can exist outside of a group). You can create, delete, and update Site Defined Groups.

Default Site Defined Groups are created automatically when a z/VM System is added to IBM Wave management. When a z/VM System is added to IBM Wave management, as part of the **Auto-Detect** process, IBM Wave classifies all of the z/VM Guests that it finds in the z/VM System. Each classification is designated its own default Site Defined Group.

Because z/VM comes preconfigured with several z/VM Guests, service machines, and others, IBM Wave can classify them in special IBM Site Defined Groups. The groups are permanently locked and it is not possible to edit, remove the definitions, or to transfer z/VM Guests in or out of the predefined groups.

Custom attributes

IBM Wave provides the **Custom Attribute Manager** that you can use to define a set of custom attributes, possible values, and a default value. The attributes can be assigned to z/VM Guests for the purpose of classifying them. You can then use the attributes in the **z/VM Guests and Groups Viewer** with the "Group By" menu option.

For example, you can define a Custom Attribute named "Client", with a default value of "Not Assigned", and the possible values "Client A", "Client B", "Client C". Next, assign several z/VM Guests the value "Client A", several with the value "Client B", and several with the value "Client C".

In the **z/VM Guests and Groups Viewer**, you can view all of the z/VM Guests in a z/VM System grouped by the "Client" attribute. Any z/VM Guest that is not assigned the "Client A", "Client B" or "Client C" value is assigned the default value of "Not Assigned".

A custom attribute must have at least one possible value and a default value assigned to it at the time of definition. The values can be changed or edited at any time. After a you define a custom attribute, all z/VM Guests are implicitly assigned the default value. This means, in essence, that all guests have some value to all defined custom attributes. In the example, z/VM Guests that did not explicitly set the value A, B, or C is assigned the default, "Not Assigned", value.

For more information, see the topic about “Custom Attribute Manager” on page 87.

Import guest metadata

The following guidelines apply when working with the **Import Guest Information** menu option.

IBM Wave provides the option to **Import Guest Information** for guest metadata that is created in a formatted comma-separated value (`.csv`) file. From the **IBM Wave Main Menu**, click **Administrative** > **Site Management** > **Tools** > **Import Guest Information**.

Use the following guidelines to create the .csv file to **Import Guest Information**:

- The imported file must be in a comma-separated value (`.csv`) format.
- The first line must contain the mandatory headers. The order of the headers is not important, but they are case-sensitive.
- The mandatory headers include:
  - `intr_username` (mandatory) - The guest name.
  - `intr_system` (mandatory) - The guest's z/VM System name.

    Note: The value for `intr_system` must be the same as the system name defined in IBM Wave.
- The optional headers include:
  - `intr_SDG` (optional) - The site defined group (SDG) name to which the guest belongs.
  - `intr_project` (optional) - The project to which the guest belongs.
  - `attr_<`*`CustomAttributeName`*`>` (optional) - The custom attribute name.
- All values that are specified in the `.csv` file must be preexisting (all SDG, Project, Custom Attribute and values, and others). If the import process locates invalid data, it generates an error. This includes guests that do not exist in the IBM Wave database.
- A blank value is ignored (attribute remains as is).

  Note: Blank values are not valid for user name or system.
- The import process generates a BTS Workunit with a BTS Request per-guest. The requests updates all the necessary fields of the guest in the IBM Wave database. You can initiate the import process and run it in the background.
- You can run an import multiple times. However, it is not possible to "undo" the changes that are made by the import process.
- The generated BTS request runs various comparisons against the database. Depending on the amount of data, the process can take a few minutes. Expect five minutes for a full update of 500 guests with all of the values entered (such as project, SDG, and four custom attributes).

For an example `.csv` file that contains the proper syntax, see Appendix B, “A sample .csv file for importing guest attributes,” on page 167.

z/VM system management

IBM Wave provides management features and options for many aspect of z/VM management. It is possible to manage storage (such as DASD, FCP, and others), networks, z/VM Guests, Linux users, and some z/VM system internal protocols.

z/VM systems and Auto-Detect

When you add a z/VM system to IBM Wave management, you can use the **Auto-Detect** process to access the z/VM LPAR and query various aspects of the z/VM system. The results of the queries populate the IBM Wave database. When **Auto-Detect** is complete, IBM Wave permits interaction with the z/VM LPAR. For more information, see “Running the Auto-Detect Wizard” on page 71.

Real device support and management

With IBM Wave, you can manage real devices visible to managed z/VM Systems. Currently, the following devices can be managed through IBM Wave:

- DASD
- OSA
- HiperSockets

Every real device that is visible to a z/VM System is assigned a **Device Pool** that owns it. Device pools can be used to allocate real devices for clone processes, or for defining new VSwitches that are connected to an Open Systems Adapter (OSA) card.

IBM Wave uses internal z/VM data to retrieve a unique ID for each real device. This unique ID is composed of the Logical Control Unit ID with other elements and provides a unique identification of the device in the installation.

For example, if z/VM System A and z/VM System B share DASD, both z/VM Systems return the same unique ID for the device. IBM Wave recognizes when a real device (DASD, for example) is dedicated to a z/VM Guest in z/VM System A, and also to another z/VM Guest in z/VM System B (both z/VM Systems must be managed through IBM Wave). This condition raises an "Attention Required" message for the real device. If the configuration is acceptable to the installation, you can ignore the "Attention Required" entry.

IBM Wave provides the function to create, modify, and update device pools. It is also possible to transfer real devices from one device pool to another.

Important: Because device pools are associated with real devices by the unique ID, all real devices that have the same unique ID are transferred to the target **Device Pool**.

A good practice is to assign real devices to device pools based on shared attributes. For example, if a site has two OSA cards, one leading to one Router 1, and another to Router 2, a good practice is to assign all real devices that are defined on the first OSA card to one device pool, and all devices that are defined on the other card to a second device pool.

Device pools are automatically created during the auto-detect processing. They use the parameters that are specified in the auto-detect process.

Important: Because device pools are associated with real devices by the unique ID, it is possible for real devices that are discovered during the auto-detection process to exist in the IBM Wave database.

Device pools are associated with z/VM Systems based on IBM Wave user specification or automatically. For example:

- z/VM System A and z/VM System B share DASD (real devices with addresses 1000-1100).
- When the auto-detection processing occurs for z/VM System A, IBM Wave links the unique ID of each device in 1000-1100 in z/VM System A with device pool "X" (that is specified by the IBM Wave User). IBM Wave also associates Device Pool "X" with z/VM System A (because it contains real devices visible to that z/VM System).

- When auto-detection processing occurs for z/VM System B, IBM Wave does not add new devices to the 1000-1100 range, but rather associates device pool "X" with z/VM System B (because it also contains real devices visible to that z/VM System).
- When you view the device pool "X" through the IBM Wave user interface. IBM Wave displays both z/VM Systems as associated with the device pool, and allows the IBM Wave user to view all that real devices that are visible to each z/VM System.

It is possible to link a **Managed Entity** to a **Device Pool**. The definition uses metadata that can facilitate a clearer and more encompassing view of the installation. For example, it is possible to link an OSA device pool to a router managed entity. The **Network Viewer** displays a link between the device pool and the router. The view indicates that the real devices owned by the device pool are connected to the router.

PAV and HyperPAV devices

If you have Parallel Access Volumes (PAV) or HyperPAV licenses in use in your z/VM environment, IBM Wave can discover only the base devices. The base devices are visible in all IBM Wave storage actions and panes. The devices are managed in the same manner as regular DASD.

IBM Wave does not display or manage the alias devices. When you perform provisioning actions on guests, such as cloning from a guest, cloning from a prototype, and duplicate z/VM definitions that have alias devices in their directory entry, IBM Wave copies the directory statement "as is" with no changes. IBM Wave does not support PAV or HyperPAV management actions on the z/VM level. All actions to manage PAV and HyperPAV devices must be done outside of IBM Wave.

z/VM guest and virtual server management

IBM Wave is designed as a central point of control with many features that can ease the management of z/VM Guests. Besides basic features for Linux Virtual Servers, like Cloning, Network connections, enhanced storage management, and other features, IBM Wave provides access methods into the z/VM Guest that run Linux.

Management assistance also comes from features like Secure Shell (SSH), 3270 and Communication-Less Connection (CLC) access, and the ability to run shell scripts on the z/VM Guests. Most management actions can be run against one or more z/VM Guests in parallel. For example, by using IBM Wave you can easily run a shell script against 20 Linux Virtual Servers in parallel.

For more information about z/VM Guest Management, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_zvm_guest_and_virtual_server_functions1.html.

Accessing Linux guests from the GUI

IBM Wave for z/VM provides multiple methods to access Linux guests from the IBM Wave user interface.

IBM Wave provides SSH, CLC, and 3270 to access a Linux guest. SSH and CLC are available provided the operating system that is running on the guest is a supported Linux version and distribution. SSH access requires user authentication by specifying either the user and password, or the private keyfile location.

SSH access can be configured in the “Security parameters” on page 125, and by “Changing User Preferences” on page 132. If the private key file contains an encrypted private key, the external SSH application is responsible for decrypting the private key with a passphrase.

For 3270 access, support is provided with or without Secure Socket Layer (SSL)/Transport Layer Security (TLS). IBM Wave supports both SSL Tunneling and Start TLS.

Note: When SSL or TLS is used, more configuration might be required.

1. If you are using SSL/TLS tunneling, the parameter must be specified in the z/VM System. For more information about the z/VM System parameters, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_display_details.html.
2. Regardless of whether SSL tunneling or Start TLS is used, either
   - The SSL/TLS certificates on the z/VM Systems must be imported into the Java keystore on the workstation.

- The "Accept All Certificates" check box must be selected in the “Functionality parameters” on page 119.

For more information about importing SSL Certificates, see Appendix J, “Configuring certificates for managed z/VM systems,” on page 191.

Deactivating Linux guests
IBM Wave provides the ability to deactivate Linux guests by using the z/VM CP commands **SIGNAL** and **FORCE**.

During the IBM Wave guest update task, which occurs on a periodic basis, IBM Wave retrieves the default value for the signal-shutdown timeout for each z/VM system. During the deactivate and recycle actions, IBM Wave uses the default value for the signal-shutdown timeout.

When no default value is set in z/VM for the signal-shutdown timeout, IBM Wave uses 32767 as the default value.

You can change the default value for the signal-shutdown timeout for a system by using the z/VM CP command **SET SIGNAL SHUTDOWNTIME**. After the value is changed in z/VM, the z/VM administrator must schedule a z/VM user update to enable the new value in IBM Wave.

For more information, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_deactivate.html.

LOGONBY support

IBM Wave supports accessing z/VM Guests by using the 3270 emulator with the **LOGONBY** statement support provided by z/VM and IBM Security Server RACF®.

When a z/VM Guest is accessed through the 3270 emulator, IBM Wave prompts for the password, and then provides access.

SSH tunneling into Linux Guests

IBM Wave does not install agents on the Linux Guests. Some of the function that is provided interacts directly with the Linux operating system that is running on the guest. For interactions, an SSH tunnel is opened between either the GUI client or Background Task Scheduler (BTS) and the Linux Guest. For authentication, IBM Wave provides several options. For more information, see “Linux Login Security Options” on page 140.

Resource verification before activation

To specify that certain checks be performed prior to z/VM guest activation, IBM Wave provides the **Check Virtual Server Resources when activating** check box on the **Administrative** > **Manage Parameters** > **Functionality** tab. The checks include:

1. Virtual Network Validity - IBM Wave checks that all virtual networks (VSwitches and guest LANs) to which the activated z/VM guest is connected exist and are operational.
2. DASD Volumes - IBM Wave performs the following checks for DASD volumes that the z/VM Guest is using:
   a. Checks that all DASD volumes used by MDISK statements in the z/VM guest CP directory entry are online and "Attached to SYSTEM".
   b. Checks that all real devices specified by MDISK DEVNO statements are online.

Note: IBM Wave does not check the following configuration options:

- DASD volumes used by the z/VM guest through the LINK statements are online and "Attached to SYSTEM".
- Dedicated devices used by the z/VM guest are online.

Functionality and Activation Levels and Activation Done signaling
Understand how to use Functionality and Activation Levels to activate and deactivate guests in a predefined order.

Using IBM Wave, you can activate and deactivate z/VM guests in a predefined order. To provide this feature, IBM Wave must be configured to be aware of the activation and deactivation order, as well as when a specific z/VM guest is considered up and running. IBM Wave uses the following components to manage the predefined order for activation and deactivation:

1. **Functionality** - A name attribute assigned to one or more z/VM guests (located in **Administrative** > **Manage Parameters** > **Functionality** in the **Functionality and Activation Levels** pane). Some examples of functionality names are database servers, web servers, and file servers (as shown in Figure 6 on page 23).
2. **Activation Level** - An attribute of the **Functionality** name. **Activation Level** indicates the activation and deactivation order for the functionality name in relation to other functionality names. For example, if Database Server A must start before Web Server A, the activation level assigned to Database Server A must be lower than the level assigned to Web Server A. If you select 10 guests with different activation levels and decide to activate with activation levels, the guests with the lowest activation level are activated first. After the guests have an active TCP/IP connection, the guests that have the next activation level are activated. The deactivate process works in reverse order. The guests with the highest activation level are deactivated first.

   In Figure 6 on page 23, notice that the database servers are at Activation Level 2 and the web servers are at Activation Level 3.

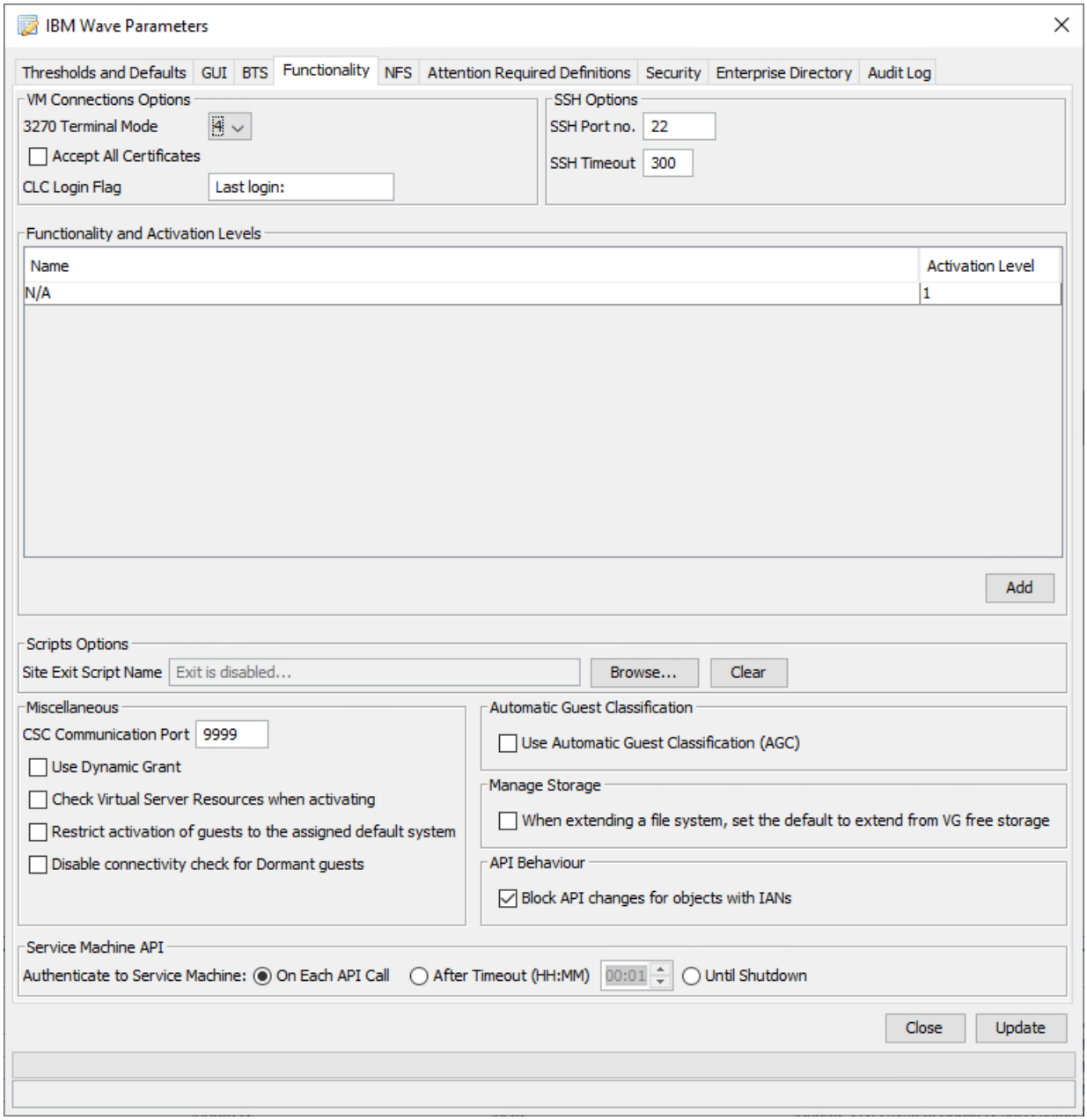


Figure 6. Functionality and activation levels: an example

3. **Activation Done signaling** - IBM Wave considers a z/VM guest to be up and running when one of the following conditions is met:

   a. The appropriate check box is selected in the Activate multiple task window and the guest has TCP/IP connectivity.

   b. A user-specified timeout value is reached.

Assign guests to a default z/VM system

When you assign the default system to one or a group of guests, you can activate all of the z/VM guests that are defaulted to the specific z/VM system in one step.

In z/VM complexes, where storage is shared across several z/VM systems, you might also want to share z/VM guests across z/VM systems. The shared z/VM guests can potentially be activated on any of the z/VM systems in the complex.

Using IBM Wave, you can define one or a group of z/VM guests to a default z/VM system. Right-click one or more guests, and select **Update** > **Assign Default System**. IBM Wave opens the window that is shown in Figure 7 on page 24.

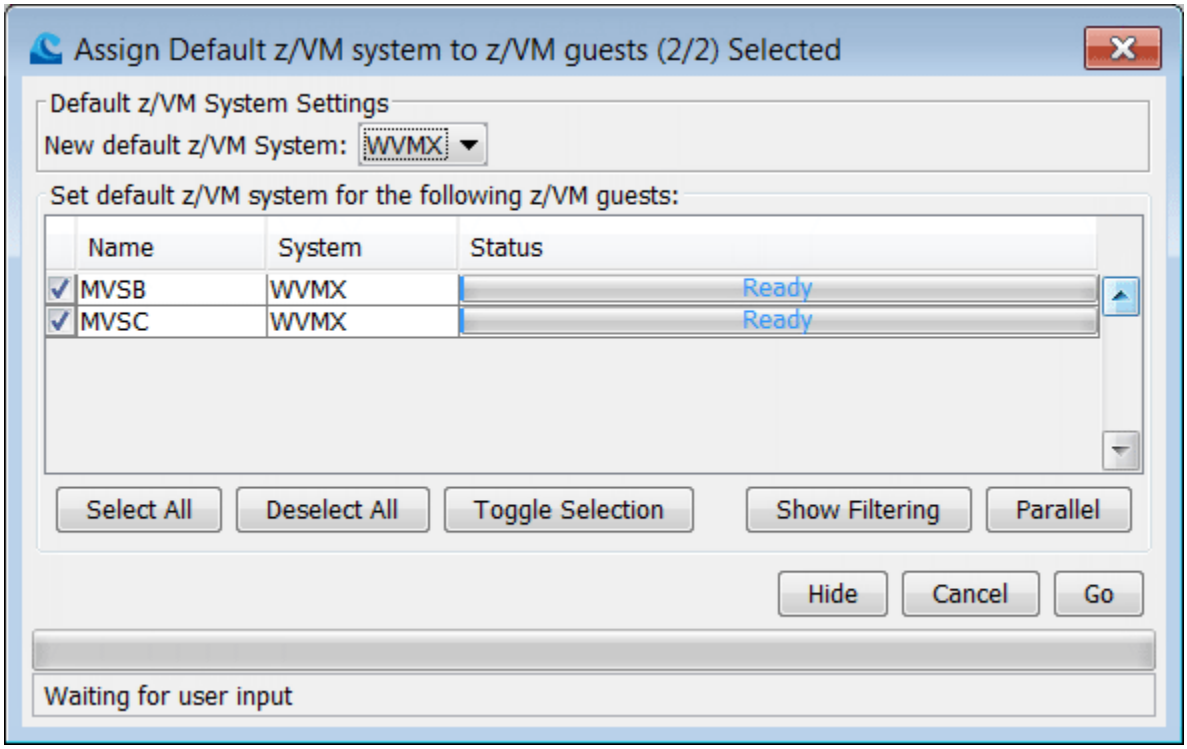


Figure 7. Assign guests to a default z/VM system

Additionally, when the default is assigned to a specific z/VM guest and someone tries to activate it on a different z/VM System, IBM Wave issues a warning message.

CLC technology
Communication-less Connection (CLC) is an IBM Wave technology that can help you solve problems when 3270 access is not available.

Communication-less Connection (CLC) is an IBM Wave patented technology that can help you solve problems. In a non-virtualized environment, when a server fails to load an operating system component, the Linux administrator can take the following actions:

1. Physically go to the server.
2. Plug in a screen and keyboard.
3. Solve the problem.

In a virtual environment, that approach is irrelevant because the server is virtual and not physical. Using CLC, you can update or change the z/VM virtual server, and edit files by using the **CLCEDIT** command.

Without CLC, the only way to solve a problem in z/VM was to open a 3270 session to the failing virtual server. IBM Wave provides standard 3270 access to the z/VM LPAR. However, 3270 access is not helpful when the applications you need to help solve the problem cannot be run.

The IBM Wave CLC technology is dynamic. Linux virtual server configuration or definitions are not needed. You can access CLC directly from the IBM Wave user interface. The only requirement to use CLC is that you can reach the z/VM LPAR through TCP/IP. When the z/VM LPAR is unreachable, CLC is not available.

Note: The vi editor is not available in CLC.

For information about how to use CLC, see CLC access.

IBM Wave verification processing

IBM Wave verifies the z/VM guest ID, host name, and the Linux operating system (OS) distribution.

IBM Wave verifies the z/VM Guest ID every time the user connects by using one of the communication methods (SSH, CLC and 3270). The process verifies the integrity of the system by comparing the z/VM Guest name of the action to the z/VM Guest ID on the virtual server, to which IBM Wave is connected.

In some cases, IBM Wave might not be aware of IP addresses changes. For example, when High Availability (HA) is in use. If virtual server A (*VSa*) and virtual server B (*VSb*) are in HA, *VSb* can instantaneously use the *VSa* IP address (or vice versa). IBM Wave becomes aware of the change when the next Network Update is run by the BTS Scheduler. If you decide to run an action on VSa, when you try to connect to VSa, you are automatically connected to VSb.

Each time a virtual server is connected to IBM Wave, the Linux OS distribution is checked. IBM Wave verifies that the OS version that is running on the virtual server is the one defined in the IBM Wave database. If IBM Wave detects an inconsistency with the OS, the z/VM Guest is placed in an inconsistent state until the matter is resolved.

IBM Wave also does a host name check to verify that the host name of the Linux virtual server is the same one that is defined in the IBM Wave database. If not, IBM Wave automatically changes the entry in the database. The change is reflected in the GUI immediately.

Initializing z/VM guests to work with IBM Wave

You must run the **Init User for IBM Wave Use** procedure on all Linux virtual servers that are managed by IBM Wave.

Although IBM Wave is agentless, it is necessary to run the **Init User for IBM Wave Use** procedure on all Linux virtual servers that are managed by IBM Wave. The initialization process verifies the existence of certain RPMs (such as CMSFS, VMCP, and others), and copies a few files into the `/usr/wave` directory. The initialization process also creates a link to the service machine minidisk for cloning purposes.

If a z/VM Guest is defined to IBM Wave as a Linux virtual server, and the **Init User for IBM Wave Use** was not run on the z/VM Guest, a warning message appears on the z/VM Guest's icon, and a warning appears in the **Attention Required** tab in the general status viewer.

When the initialization for IBM Wave process runs against a Linux Guest, IBM Wave records and stores the successful and failed parts of the initialization process. The records can be used whenever IBM Wave needs to assess whether a particular action is valid for a specific guest. For example, when a certain guest is configured to use the by-UUID method of referencing DASD volumes in its `fstab` or `zipl`, any cloning action is disabled ("Clone", "Convert to Prototype", and others). However, actions such as "Execute Script" and "Connect/Disconnect to/from VNS" are valid for the guest.

The **Attention Required** entry for the guest contains a list of elements for which the **Init User for IBM Wave Use** action failed. The tooltip for the z/VM Guest contains the same data.

During the initialization process for a Linux guest, the Linux OS distribution release is verified for full or partial support. Some distributions might be only partially supported by IBM Wave for z/VM. Partially supported guests do not have all of the available guest actions enabled. For example, the "Manage Storage" action might be disabled for a partially supported guest, and a tooltip explains the reason. For a list of supported actions, see Appendix A, “Linux distribution support,” on page 165.

For more information, see Init Users for IBM Wave use.

Bare-metal installation (BMI) support

The Bare-Metal Installation Wizard helps simplify the installation of a new Linux operating system on a z/VM guest, which can be helpful for personnel with no z/VM background.

For more information, see Installing Linux with the BMI wizard.

Notes:

- For Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), and Ubuntu, make sure the installation guest is connecting through an access port type connection. Currently, Ubuntu, RHEL, and SLES cannot install when connected by using the trunk port type.

- During the interactive installation phase for Ubuntu, it is recommended that the Basic Ubuntu Server is included during the software selection phase. Doing so ensures that most of the packages that IBM Wave needs are automatically installed on the server, and the different Linux actions can work properly.

For more information, see Connecting z/VM guests to virtual network segments.

Mechanism

Installing a new Linux operating system on a z/VM guest can often be a challenge for personnel without a z/VM background. The task includes the following actions.

1. Building a Linux PARMFILE that includes various parameters for the original installation program. The parameters include the host name configuration, password configuration, and networking related parameters (IP address, DNS, default gateway, and more).
2. Copying the files for starting the installation program to the target z/VM guest's reader.
3. Starting the guest, and then starting the original installation program.

IBM Wave stores all of the files relevant to a Linux installation on a dedicated minidisk on the short IBM Wave service machine. This minidisk is created automatically during the first use of a Linux repository on each z/VM system (because the IBM Wave service machines are specific for each z/VM system). The following files are stored on the dedicated minidisk:

- The Linux VM kernel that is used by the local installation program (named `VMRDR.IKR` or `KERNEL.IMG`).
- The Linux `INITRD` file (named `INITRD` or `INITRD.IMG`).
- The PARMFILEs that are generated by the **Launch Linux Installation** action when you are using the Linux repository.

IBM Wave uses multi-write (MW) disks for the dedicated minidisk for BMI. To avoid any possible disk corruption issues, one of the following courses of action should be taken:

1. Avoid performing bare-metal installations (BMIs) on different members of the SSI at the same time.
2. Create different Linux repository definitions in the IBM Wave Linux Repository Manager for each system in the SSI or shared directory.
3. Pre-define the 4xx disks (starting with 400) using DirMaint commands in each of the subconfigurations for an SSI or SYSAFFIN statement of a shared directory.

The first time the Linux media repository uses the "Launch Linux Installation" process, the dedicated minidisk is created and the Linux installation files are sent to it (by using FTP). During subsequent **Launch Linux Installation** actions, the Linux installation files are verified, and then sent by using FTP again, if necessary.

The building of the PARMFILE is done according to the parameters supplied by the user who runs the **Launch Linux Installation** action. This PARMFILE is then sent by using FTP to the dedicated minidisk on the IBM Wave Service machine.

After you submit the bare-metal installation (BMI) request, IBM Wave pushes all necessary files into the target z/VM guest's reader and starts the z/VM guest.

After the z/VM guest is started, and the installation program loads, IBM Wave samples the supplied IP address periodically until the SSH port is available. When the SSH port is ready, a message is displayed to the person who started the installation. Now, you can open an SSH console into the z/VM guest, and start the installation program.

During the installation process, IBM Wave continues to monitor the progress of the installation by displaying a progress bar under the z/VM guest icon. After the installation program is done, the installation status of the z/VM guest reflects completion.

The target z/VM guest icon reflects the status of the Linux installation, which can include one of the following statuses:

- During the **Initializing Interactive Installation** stage, the service machines build the PARMFILE and push the Linux installation files to the z/VM guest's reader. During the last phase, the z/VM guest is started from the reader.

- The **Interactive Installation Ready** status indicates that the Linux installation reached a stage where the user must connect to the z/VM guest (by using SSH) to start the **Launch Linux Installation**.
- The **Interactive Installation In Progress** status indicates that the Linux installation program is started by the user. IBM Wave continues to monitor the progress of the installation by using the console output from the z/VM guest.
- The **Interactive Installation Complete** status indicates that the Linux installation program completed the installation and the z/VM guest is now running the installed Linux operating system (OS).
- The **Interactive Installation Status Unknown** status indicates that an interactive installation was detected during BTS initialization, but IBM Wave is not tracking the installation status. Therefore, the status of the installation is unknown.
- The **Interactive Installation Failed** status indicates that the Linux installation program did not successfully start. For more information about the error, review the COR output of the **Launch Linux Installation** BTS request.

z/VM guest profile support

IBM Wave takes the z/VM Directory statements into account, which appear in the linked z/VM Profile for a z/VM Guest. For example, if a NICDEF statement appears inside a z/VM Profile, and the z/VM Profile is linked to a z/VM Guest, IBM Wave shows it in the **Network Viewer**.

In addition, IBM Wave detects virtual devices that appear in the linked z/VM Profile when it attempts to add a virtual address to a z/VM Guest. (For example, when adding Storage to a z/VM Guest, connecting to a Virtual Network Segment, and other scenarios.)

Storage management

IBM Wave helps to simplify viewing storage status with the following interfaces:

- “DASD storage status” on page 27, and DIRMAINT regions.
- “FCP-attached storage” on page 28.

DASD storage status

Use the **Storage Viewer** to view many aspects of storage status.

To view DASD status in the **Storage Viewer**, from the **Hardware Viewer** select the **Current System Viewer** > **Storage Viewer**.

IBM Wave assigns the following types of status to each DASD Volume that it manages:

- **Enabled Storage** - Volumes that are recognized by the z/VM System, but are not managed at all by the directory manager.
- **Defined Storage** - Volumes that have a defined DIRMAINT region.
- **Assigned Storage** - Volumes that are managed by the directory manager and are assigned to a DASD group (DIRMAINT) or subpool.
- **Page** - Volumes that are marked as page volumes.
- **Spool** - Volumes that are marked as spool volumes.
- **Page + Spool** - Volumes that consist of both a page space and spool space.
- **User Attached** - Volumes that are attached to specified z/VM Guest.
- **Free** - Volumes that are online, not attached to a system or a user, and not owned by a CPC.
- **CP Owned** - Volumes that are attached to system and are not owned by a CPC.

DIRMAINT does not allow the creation of empty storage groups. To ease the creation of a new storage group without needing to immediately assign a DIRMAINT region to it, a "Dummy Region" is created. The "Dummy Region" is one cylinder in size. It is created during the **Auto-Detect** process when you add a z/VM System to IBM Wave management (if the directory manager on that z/VM System is DIRMAINT). Although this region must be defined on a real DASD Volume, it is never physically allocated and is used as a placeholder for empty storage groups.

Restriction: Never define the "Dummy Region" larger than one cylinder.

Notes:

- IBM Wave issues interface messages to notify you about storage inconsistencies.
- When there is an issue with a DASD volume, it is indicated with an attention required flag.
- When there is an issue with a DASD volume that is associated with the DASD group, you can disable the DASD volume group.

FCP-attached storage
IBM Wave provides tools to assist you with the management of FCP-attached storage.

FCP support is divided into the following categories:

1. **Inventory Management** - IBM Wave can detect all FCP devices visible to the z/VM System, and assign them to a default Device Pool. IBM Wave keeps track of FCP devices that are assigned to z/VM Guests (either permanently using the **DEDICATE** directory statement, or online by using the **ATTACH CP** command). Any action that requires the dedication of a new FCP device automatically provides only the available FCP devices.

   By default, every Target World Wide Port Name (WWPN) detected by IBM Wave is assigned to a default storage controller. Target World Wide Port Names (WWPNs) can be associated with other storage controllers by selecting **Administrative** > **Site Management** > **Manage FCP Information**. See “Manage FCP Information” on page 94.

   All of the data, in conjunction with other information is used to generate the Storage Layout view for a specific z/VM guest or a group of guests. For more information, see Generate Disk Storage Map.
2. **Cloning** - When you clone a guest, you can request that new FCP devices be assigned to the target clones. The new FCP devices are retrieved from the eligible FCP device pools and are dedicated to the new guests using the **DEDICATE** directory statement. For more information, see Clone.
3. **Manage Storage** - FCP support is fully integrated in the **Manage Storage** feature of IBM Wave. You can attach new FCP devices and new LUNs to Linux guests. For more information, see Manage Storage.

Prototype management

Prototypes (or skeletons in VM: Secure/VM:Direct) are directory manager entries that can be easily cloned from. A prototype is a definition of a z/VM guest. IBM Wave takes the prototype concept to a new level.

Note: Although the terminology varies between directory managers, a *prototype* in this document refers to DirMaint prototypes and equivalent skeletons.

Instead of simply using prototypes for physical cloning, IBM Wave provides the means to clone the physical data and the logical data of a server.

There are two types of prototypes in IBM Wave:

1. Associated prototypes
2. Dissociated prototypes.

Associated prototypes are prototypes that have a z/VM guest assigned to them. When cloning from such a prototype, the IBM Wave user has two choices:

1. **Duplicate this prototype's user definition** - Create a new z/VM guest definition according to the definition of the prototype.

   This process is also referred to as *physical cloning*, or making a shallow copy.
2. **Clone from this prototype** - Create a new z/VM guest definition and copy the data from the assigned user to the newly-created user.

   This process is also referred to as *full cloning*, or making a deep copy.

Dissociated prototypes do not have a z/VM guest associated with them. When cloning from a dissociated prototype, the user can only choose physical cloning. Also, dissociated prototypes are only visible to IBM Wave users with the Site Level Administrator role.

The IBM Wave user can create new prototypes, associate z/VM guests with prototypes, delete prototypes and more, provided the corresponding permissions are granted.

When converting a z/VM guest into a prototype, that z/VM guest is removed from the display, so no action can be done against it. Also, IBM Wave will change that z/VM guest's z/VM password to NOLOG in order to avoid accidental activation of this z/VM guest. During association of a z/VM guest to a prototype, a verification process is run in order to make sure that the z/VM guest's z/VM definitions match those of the z/VM prototype.

Prototypes are managed from the **Prototypes Viewer** in the **Current System Viewer**.

Network management

IBM Wave simplifies network management in the managed z/VM systems. IBM Wave provides functionality to create, update, and remove virtual networks (guest LANs and VSwitches), and connect and disconnect virtual guests to and from virtual networks.

Virtual networks

The term *virtual network* is a term that is used to describe either a Guest local area network (LAN)[3] or a VSwitch. IBM Wave automatically detects every defined virtual network during the auto-detect process. IBM Wave contains options to create, update, and remove virtual networks (one or more Guest LANs and VSwitches). When you create VSwitches by using IBM Wave, you can define a triplet of OSA devices through which the VSwitch connects to the OSA card.

Dynamic and static GRANT processing

When a new connection is defined from a z/VM Guest to a VSwitch, by using the "Magic Wand" or when cloning z/VM Guests, IBM Wave automatically issues **GRANT** commands to permit z/VM Guests to connect to the VSwitch. IBM Wave provides two ways of managing the **GRANT** commands:

1. **Dynamic**: When using the dynamic method, instead of adding the GRANT command to the GRNTPROF file, IBM Wave issues the GRANT command dynamically as part of the Activate Action.
2. **Static**: When using the static method, IBM Wave adds a GRANT command to the GRNTPROF file in the AUTOLOG machine. You can specify the GRANT processing method in the IBM Wave parameters.

Note: IBM Wave has limited support for switching between static and dynamic modes. When switching from dynamic to static, IBM Wave cannot add the necessary GRANT commands to the GRNTPROF AUTOLOG file.

Related topics:

- "Making VSwitches permanent" on page 29
- "Configuring AUTOLOG" on page 35
- "Functionality parameters" on page 119

Making VSwitches permanent

To make a VSwitch permanent, add the VSwitch definition to the **LANPROF** file.

Before you begin

When you use the **Auto-Detect** process on a z/VM system, the existing VSwitches are treated as persistent. The persistent state causes any GRANT created by IBM Wave for the existing VSwitches not to be added to the **GRNTPROF** file. Because the VSwitches are not in the **GRNTPROF** file, during the next IPL when the **GRNTPROF** file is run, the GRANTS are not automatically created.

[3] A virtual local area network (LAN) segment that is emulated by the z/VM Control Program (CP). A Guest LAN can be shared by guest virtual machines on the same z/VM system.

About this task

To make any VSwitches that are not created by IBM Wave permanent, take the following steps.

Procedure

1. To make VSwitches permanent, edit the **LANPROF WAVEPARM** file.
2. Add the VSwitch definitions.
3. Refresh IBM Wave by using the **Schedule z/VM Network Update**. From the **IBM Wave Main Menu**, select **Auto Detect** > **Refresh** > **Schedule z/VM Network Update**.

Results

The VSwitches are now permanent in IBM Wave.

Note: If you must make a VSwitch persistent, remove the VSwitch statement from the **LANPROF WAVEPARM** file.

Related topics:

- “Configuring AUTOLOG” on page 35
- “Dynamic and static GRANT processing” on page 29

Review the parameter files

This task applies to each managed z/VM system.

About this task

Every time an IBM Wave user connects a z/VM guest to a permanent VSwitch, a **GRANT** command is added to the **GRNTPROF** parameter file. The **GRNTPROF** file is run during an IPL, and all the grants created by IBM Wave are issued.

Any VSwitch definitions that are not created by IBM Wave are considered *persistent*. Any grants that are issued to persistent VSwitch definitions are not placed in the **GRNTPROF** file and cannot be issued after an IPL.

If you want IBM Wave to treat the VSwitches as "permanent", you must add a definition statement for the VSwitch in the **LANPROF WAVEPARM** file.

Note: You must also add a definition statement for any future VSwitches that are created outside of IBM Wave for z/VM.

Procedure

1. Review “Dynamic and static GRANT processing” on page 29 and “Making VSwitches permanent” on page 29.
2. Adjust the **LANPROF WAVEPARM** file.

Virtual network segment

The term *Virtual Network Segment (VNS)* refers to a logical definition within IBM Wave that defines an IP Network segment. A Virtual Network Segment contains the following IP network definitions:

- **Network** - The network segment (For example 198.51.20.0).
- **Netmask** - The netmask for the segment (For example (255.255.255.0).
- **Broadcast** - The broadcast of the IP segment (For example (198.51.20.255).
- **Default Gateway** - An address on the network segment that is used as a default gateway when z/VM Guests connect to the VNS (for example 198.51.20.254).
- **Default NIC Address** - A virtual NIC address that is used to connect z/VM Guests to the Virtual Networks. For example, 1F00. When a Guest is connected to a Virtual Network through a VNS, the

default NIC statement in the guest user directory specifies 1F00 as the virtual NIC address, and all Linux configuration files refer to the 1F00 virtual address.

- **VLAN ID** - An optional VLAN ID to be associated with this VNS. The VLAN ID is used when the z/VM Guest is connected to the VNS.
- **BTS Enabled** - A check box that indicates whether the Background Task Scheduler (BTS) can attempt to use connections routed through a VNS to connect to guests. When the check box is selected, the BTS attempts to use IP addresses from the VNS to perform managed guest actions (such as querying performance data or managing storage). When the check box is not selected, the BTS ignores any IP addresses that come from the VNS, and cannot use it to connect to guests.

The VNS definitions are used by IBM Wave to connect z/VM Guests to Virtual Networks. There is a many-to-many relationship between the VNS and the Virtual Network (multiple Virtual Network Segments can be associated with one Virtual Network, and multiple Virtual Networks can be associated with one Virtual Network Segment). IBM Wave automatically creates VNS definitions for existing IP connections to existing Virtual Networks during the **Auto-Detect** process.

For example, if z/VM Guest A is connected to VSwitch 1 with IP address 198.51.20.30, and z/VM Guest B is connected to VSwitch 2 with IP address 198.51.30.30, IBM Wave automatically creates two VNS definitions:

- VNS A with network 198.51.20.0
- VNS B with network 198.51.30.0

IBM Wave automatically updates its knowledge base that VNS A is associated with VSwitch 1, and VNS B is associated with VSwitch 2. IBM Wave also updates its database that Guest A is connected to VNS A with IP address 198.51.20.30, and Guest B is connected to VNS B with IP address 198.51.30.30.

Note: This information is available in the **Network Viewer** when you select the corresponding z/VM System.

Whenever IBM Wave detects a Virtual Network in the z/VM System, a special "Unknown IP" VNS is created and is associated with the Virtual Network. This VNS is a special VNS that cannot be removed, updated, or associated with other Virtual Networks.

IBM Wave allows a VLAN ID to be defined as part of the VNS definition. The VLAN ID is used when z/VM Guests connect to the VNS.

When an IBM Wave User attempts to connect a z/VM Guest to a VNS, the next available IP address that is not in use is assigned to the Guest connection.

When a VNS has a VLAN ID defined, it is also possible to connect an IBM Wave Managed Entity to the VNS as metadata to indicate that the VLAN is defined on a specific router. The Network topology diagram that is shown in the **Network Viewer** for a z/VM System shows the relationship.

VNSs are defined as global, per IBM Wave Server. Different z/VM Systems can use the same VNS for z/VM Guest connections. This means that when IBM Wave suggests an available IP address, it is not known to be used in any z/VM System managed by IBM Wave. Furthermore, if different z/VM Systems use the same VNS, it is illustrated in the Network topology diagram.

The following examples to illustrate some of the Virtual Networks and Virtual Network Segments concepts:

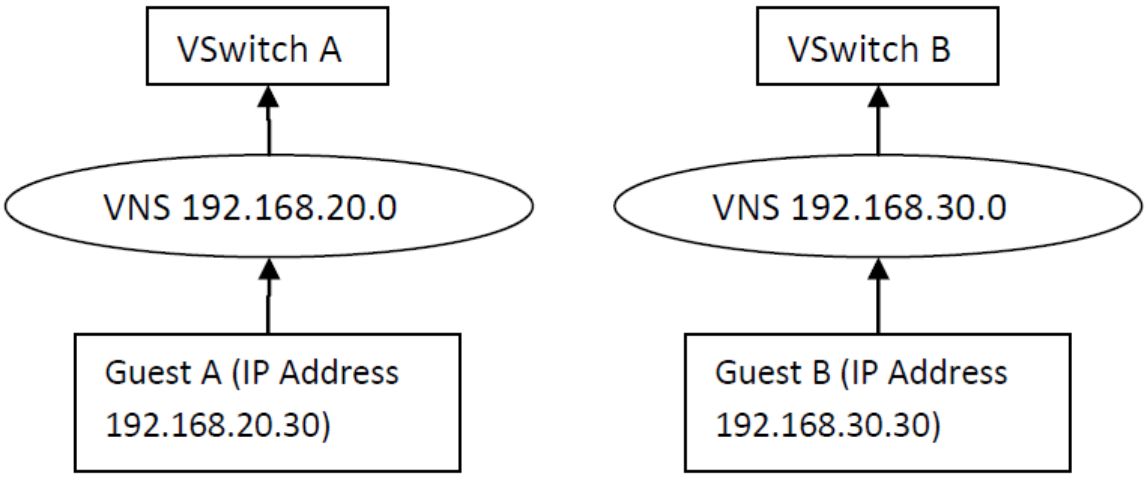


Figure 8. Two VSwitches with a guest on each one with different IP address segments

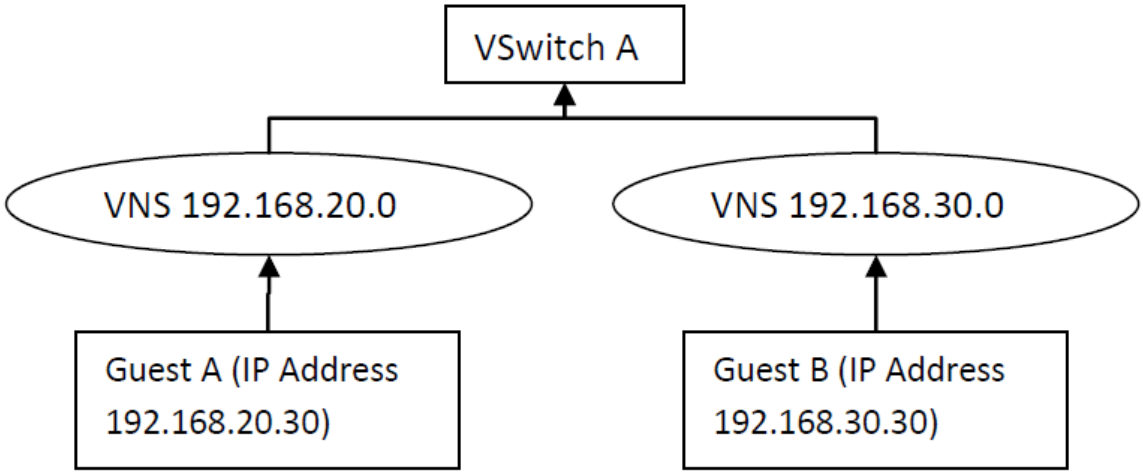


Figure 9. One VSwitch routing two IP network segments

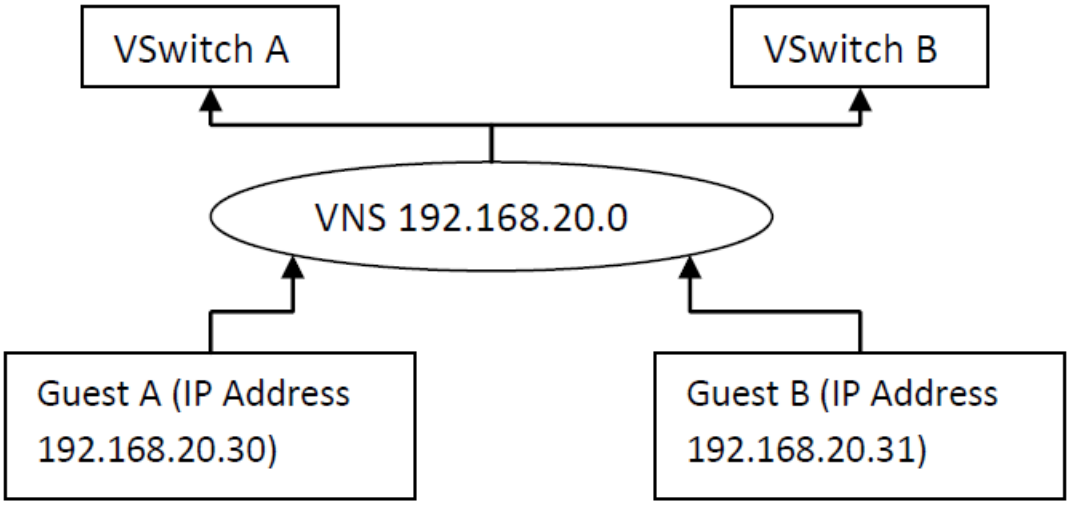


Figure 10. Two VSwitches routing the same IP segment

Notes:

1. IBM Wave automatically detects the IP addresses of z/VM Guests that are active and connected to a Virtual Network. Guests that are not active, yet contain a default NIC statement that connects them to a specific Virtual Network are detected and connected to an Unknown IP VNS that is associated with the Virtual Network.
2. When a VNS is automatically created based on IP addresses of connected z/VM Guests, IBM Wave attempts to put the most logical information in the VNS definitions, based on the IP address of the Guest. For example, a Guest that is connected to a Virtual network with IP address 198.51.20.30 triggers the creation of a VNS whose network is 198.51.20.0, broadcast 198.51.20.255, netmask 255.255.255.0 and default gateway 198.51.20.254. These parameters can be altered through the **Update Information** action for the specific VNS.
3. IBM Wave is unable to automatically determine whether a z/VM Guest connection is using VLAN tagging. This information is retrieved from the VNS, where it is set as an attribute by the IBM Wave administrator as metadata field.

For more information, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_virtual_network_segment_functions1.html.

Interactions with the TCP/IP virtual machine

The Transmission Control Protocol/Internet Protocol (TCP/IP) router is the z/VM TCP/IP configuration, or another virtual configuration in the z/VM System that runs the z/VM TCP/IP stack.

Note: This information does not describe routing a Linux configuration.

A Guest LAN has no outside interface. z/VM Guests that are connected to the Guest LAN must be routed through the z/VM system TCP/IP stack. Or they can be routed through another z/VM Guest that is running a TCP/IP stack and acts as the router.

IBM Wave simplifies the task of "routing a Guest LAN" by allowing an IBM Wave user to define a TCP/IP router to each defined Guest LAN. If you decide to route the Guest LAN through TCP/IP, IBM Wave makes the following changes to the TCP/IP router:

1. Dynamically connects the TCP/IP router z/VM Guest to the Guest LAN.
2. Adds the following items to the PROFILE TCP/IP parameter file:
   a. Device and Link statements that connect to that Guest LAN.
   b. Gateway statement (according to the network, netmask, and broadcast parameters specified in the Guest LAN definition).
   c. Home statement (according to the Default Gateway parameter specified in the Guest LAN).

Notes about multiple IP addresses over the same network interface

IBM Wave provides special handling in cases where a Linux Guest uses multiple IP addresses per NIC. The use of multiple IP addresses can occur in the following situations:

- The Linux Guest is connected through a trunk connection over one (or more) of its network interfaces.
- Several IP addresses are defined on the same NIC.

During **Auto-Detect** processing, IBM Wave recognizes the situation and reflects it in the **Network Viewer**.

z/VM page and spool disk management

IBM Wave allows an IBM Wave administrator to add Page and Spool disks to a z/VM System. This process depends on the following factors:

1. The Page/Spool disks must be pre-formatted with a Page/Spool space.
2. The disks must be marked as Page or Spool through the Storage Viewer.
3. A CP slot for the disks must be available.
4. If the z/VM System is part of an SSI complex, the volume must be properly formatted with the SSI cluster name.

Through the "Mark as Page" or "Mark as Spool" DASD volume actions, you can mark a DASD volume for the "Add Page" or "Add Spool" z/VM system actions.

For more information about these actions, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_zvm_system_actions.html and https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_dasd_volume_functions.html.

z/VM utilization and performance statistics

IBM Wave provides performance and utilization statistics in the **Hardware Viewer** > **System Status**. The statistics include:

- CPU and paging Statistics for the entire z/VM system
- CPU and other performance statistics of specific z/VM guests running in the z/VM system
- Virtual to real storage ratio
- Page space utilization
- Spool space utilization.

The utilization statistics are gathered by the BTS and stored in the IBM Wave database. The information is displayed as dial plots in the **System Status** viewer. You can drill down into each of these dial plot to see more data.

You can specify the method through which performance data is retrieved for each z/VM system that is managed by IBM Wave. IBM Wave can retrieve performance data by using the z/VM **INDICATE** command, or Performance Toolkit for z/VM.

When performance data is retrieved by using the **INDICATE** command, CPU and performance data is not available for each z/VM guest.

When performance data is retrieved through Performance Toolkit for z/VM, you can see detailed performance data for every active z/VM guest, and globally for the z/VM system. CPU utilization for z/VM guests that are not in the user's scope are aggregated in the bottom row and are shown as `NOT IN SCOPE USERS`.

Note: z/VM guest CPU usage percentage is calculated as follows:

- If the z/VM guest has one virtual CPU, the percentage will be the raw data retrieved from Performance Toolkit for z/VM.
- If the z/VM guest has two or more virtual CPUs, the percentage of CPU used is displayed as normalized while considering the number of virtual CPUs defined to the guest. For example, if a guest has CPU1 at 20%, CPU2 at 30% and CPU3 at 10%, the total CPU utilization will show

  (20 + 30 + 10) / 3 = 20%.

Inconsistency mechanism

When the BTS detects a z/VM object that has been deleted outside of IBM Wave (manually by a z/VM administrator, for example), it places that object in a state of "inconsistency". This state is reflected in the icon of that object, and signifies that the object no longer exists in the z/VM system.

Because IBM Wave keeps extensive information on certain z/VM objects, it is sometimes possible to "restore" a deleted object. If a z/VM administrator deleted a guest LAN for example, the BTS will mark that guest LAN as "Inconsistent". This will be reflected in the IBM Wave client. The IBM Wave user can then select "recreate GLAN" from the popup menu of that guest LAN.

There are various types of inconsistency, depending on the z/VM object.

It is also possible to delete the inconsistent object from the IBM Wave database.

Cross-system cloning and minidisk-streaming process

The following topic explains the cross-system cloning and minidisk-streaming process.

IBM Wave provides functions to clone z/VM virtual guests across IBM Wave managed z/VM Systems. This process is initiated from the clone action and can be done on either z/VM Guests or z/VM Prototypes.

Cross-system cloning (CSC), like other clone processes are done in two phases. The first phase is generally the same as in a regular clone process.

1. In the first phase, the new z/VM Guest and network definitions are defined on the target z/VM System, which is different from the source z/VM System.
2. In the second phase, the Background Task Scheduler (BTS) clones the minidisks of the source z/VM Guest by using minidisk-streaming.

The minidisk-streaming process transmits data on a minidisk in the source z/VM System through TCP/IP to the target z/VM System by using the cross-system cloning service machine (WAVEWRKC). Therefore, the process is serial, and if the source z/VM Guest has three minidisks, they are streamed one after the other. You can start any number of minidisks and concurrent CSC processes. However, each minidisk-streaming process uses a BTS worker (one BTS worker per minidisk-streaming process).

For example, if a source of a cross-system cloning process is a virtual guest with three minidisks, three BTS workers are allocated to the process. Also, minidisk-streaming is done through the WAVEWRKC service machine and is done serially. The minidisk-streaming process occupies a BTS worker before it is handled by the WAVEWRKC service machine and while it is waiting on the cross-system cloning work queue. Therefore, the recommendation for cross-system cloning, is to define at least two BTS workers per managed system. Depending on the network speed, average number of minidisks for z/VM Guests that are targeted as cross-system clone sources, add between two and six BTS workers per managed z/VM System.

For more information about cross-system cloning, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_clone_from_this_prototype.html.

Configuring AUTOLOG

AUTOLOG can be configured to the needs of your installation.

When z/VM starts, it can be configured to automatically activate the AUTOLOG1 guest. Typically, when you use an external security manager (ESM) in your environment, AUTOLOG1 activates the ESM, which then starts the AUTOLOG2 guest. Using this logic, z/VM ensures that the security product, such as RACF, is running before any other programs or commands.

Both AUTOLOG1 and AUTOLOG2 search for the PROFILE EXEC file to run it. To activate the IBM Wave service machines, you must add the following line to the PROFILE EXEC file:

```
'EXEC WAVEAUTR ACTPROF'
```

Note: The TCP/IP stack must be activated before the IBM Wave service machines.

To use the AUTOLOG1 and AUTOLOG2 machines for LAN and EDEV definitions, add the following three lines before activating guests that use LAN or EDEV resources:

```
'EXEC WAVEAUTR EDEVPROF'
'EXEC WAVEAUTR LANPROF'
'EXEC WAVEAUTR GRNTPROF'
```

The following WAVEPARM definitions are provided for each file:

ACTPROF WAVEPARM
The ACTPROF WAVEPARM file contains the commands to start the service machines. The default service machine names are WAVEWRKS, WAVEWRKL, and WAVEWRKC.

EDEVPROF WAVEPARM
The EDEVPROF WAVEPARM file contains commands that define the emulated devices (EDEV) and its SCSI path. The file is updated when an EDEV is created, modified, or deleted. The EDEVPROF WAVEPARM file contains the z/VM CP commands that created the EDEV and associated paths. If the EDEV was formatted as part of the creation process, the commands to vary the EDEV online and attach it to the system are also listed.

LANPROF WAVEPARM
The LANPROF WAVEPARM file contains definition statements and commands for the VSwitches and Guest LANs. Every time an IBM Wave user creates a permanent Virtual Network, a definition statement is added to the LANPROF file. When the Virtual Network is deleted through IBM Wave, the entry is also deleted from the LANPROF file.

GRNTPROF WAVEPARM
The GRNTPROF WAVEPARM file contains the **GRANT** commands for each VSwitch and restricted Guest LAN. Each time an IBM Wave user connects a z/VM Guest to a permanent VSwitch, a **GRANT** command is added to the GRNTPROF parameter file to enable the guest to remain connected after an IPL. When a user is disconnected, or when the VSwitch is deleted, the relevant entries are deleted from the GRNTPROF file.

Related information

"Installing and customizing IBM Wave" on page 51
z/VM: CP Planning and Administration for all z/VM releases

Directory manager generated work units

Some actions that use z/VM management can take longer to run than others. For example, copying disks and deleting z/VM guests. IBM Wave uses standard System Management API (SMAPI) to make updates and query to and from a z/VM System (including long-running tasks).

Some Systems Management API calls can cause the directory manager to create a work unit for the task. The work units are managed by the directory manager and are run by using a directory manager service machine. The decision whether to run a task in a work unit or ad hoc is done by the directory manager.

After Systems Management API calls, running in a Background Task Scheduler (BTS) request creates a work unit, the internal z/VM work unit sampler component in the BTS begins tracking it. The COR output for the BTS request specifies the ID given to the work unit by the directory manager and its progress, which is based on periodic sampling.

Note: Work unit sampling occurs only when active work units are running in the z/VM System. The sampling interval for the work units is 30 seconds by default, but can be modified in the **Administrative** > **Manage Parameters** > **IBM Wave Parameters** window.

z/VM account management

An installation can use z/VM account information to classify and categorize z/VM guests for various purposes such as monitoring performance, reporting, billing, and other tasks.

IBM Wave automatically detects z/VM account information that is used in managed z/VM systems and stores that information in the IBM Wave database.

To manage z/VM account information (view, add, update and remove), use the **Administrative** > **Site Management** > **z/VM Account Manager** menu option. For more information about the **z/VM Account Manager**, see Chapter 4, "Administrative actions," on page 85.

To assign a new account or remove an existing account definition for one or more z/VM guests, you can use the **Assign Account** or the **Remove Account** multiple task action. For more information, see:

https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_zvm_guest_and_virtual_server_functions1.html

In the **z/VM Users** tab and in the **Enterprise Viewer**, z/VM guests can also be grouped by or filtered by the z/VM account.

Note: IBM Wave supports assigning a primary account number to z/VM guests. Secondary account numbers and distribution codes cannot be assigned using IBM Wave, but if they are assigned outside of IBM Wave, they are unchanged when you use IBM Wave to assign primary account information.

Automatic Guest Classification

Use **Automatic Guest Classification** to associate IBM Wave metadata elements with the z/VM Guest's directory entry.

Automatic Guest Classification (AGC) can analyze changes to the z/VM Guest's directory entry. You can configure AGC to do the following tasks automatically:

- Assign IBM Wave metadata elements to a z/VM Guest based on an analysis of the z/VM Guest's directory entry.
- Change the z/VM Guest's directory entry after a user-driven change is made to the metadata assigned to the guest.

Each AGC entry is associated with an **AGC Property**. AGC queries the value for the Guest's **AGC Property**, and defines the **AGC Property** as the primary z/VM account value.

If the guest's directory entry has an INCLUDE statement, AGC searches the included profile for an ACCOUNT statement. However, the guest's primary account value for the ACCOUNT statement takes precedence over any ACCOUNT statement in a profile referenced by an INCLUDE statement. For an example of the analysis results, see row 4 in Table 3 on page 37.

Table 3. Analysis when querying the AGC property value

Account Statement for the z/VM Guests' Directory Entry	**Analysis Results**
ACCOUNT **USER1**	**USER1**
ACCOUNT **USER1** USER2	**USER1**
ACCOUNT **ACCT3** USER1 ACCT1 ACCT2	**ACCT3**
INCLUDE PROF1 and PROF1 contains ACCOUNT **B5DD7** B5DD8 B6DD7 B5DD9	**B5DD7**
No Account statement	*<AGC Property Not Found>*

When used to set the AGC Property value, AGC sets the primary account value for the z/VM Guest. Table 4 on page 37 shows examples of the "**Set**" results.

Table 4. Setting the value for the AGC property

Account Statement before "Set"	**Account Statement after "Set" with AGC property value "ACCTSET"**
ACCOUNT USER1	ACCOUNT **ACCTSET**
ACCOUNT USER1 USER2	ACCOUNT **ACCTSET USER2**
ACCOUNT ACCT3 USER1 ACCT1 ACCT2	ACCOUNT **ACCTSET USER1 ACCT1 ACCT2**
INCLUDE PROF1 and PROF1 contains ACCOUNT B5DD7 B5DD8 B6DD7 B5DD9	ACCOUNT **ACCTSET B5DD8 B6DD7 B5DD9**

Related information

"AGC Manager" on page 37

AGC Manager

Use the Automatic Guest Classification (AGC) Manager to create, update, and delete AGC Entries.

To access the **AGC Manager**, click **Administrative** > **Site Management** > **AGC Manager**.

Important: After AGC is enabled in the **IBM Wave Parameters**, the AGC rules are strictly enforced. For parameter information, see "Running Automatic Guest Classification (AGC)" on page 43.

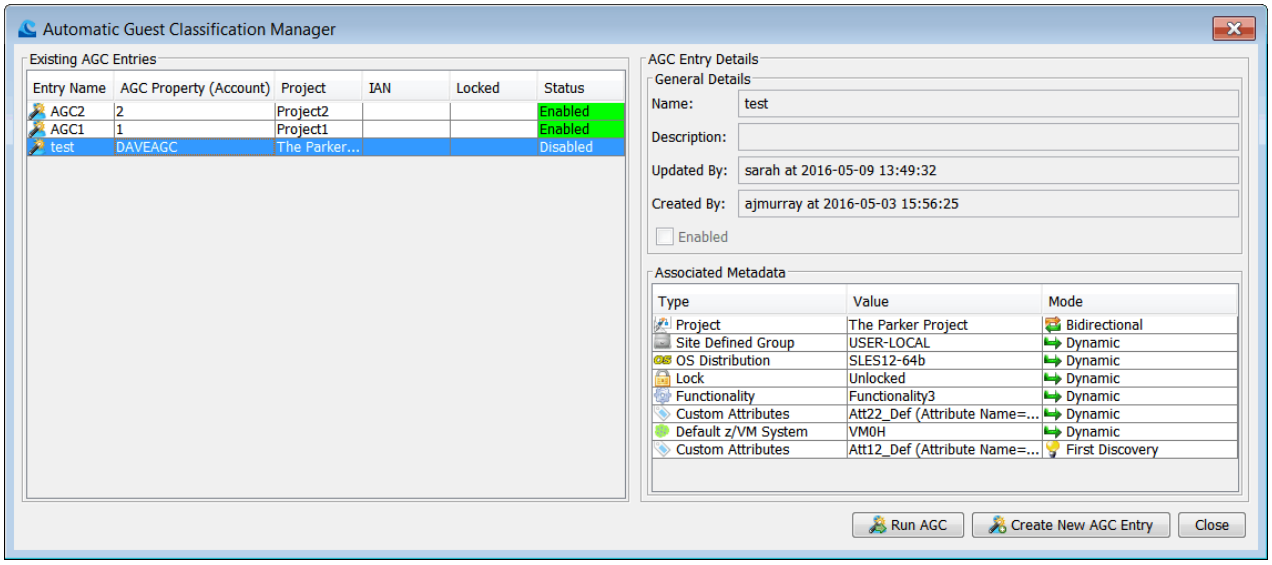


Figure 11. Automatic Guest Classification Manager

The **Existing AGC Entries** pane contains a table with all of the AGC entries that are defined in IBM Wave. The following columns are in the table:

Entry Name
The AGC entry name.

AGC Property (Account)
The **AGC Property** is the primary z/VM ACCOUNT value for the **Entry Name**. The title of the cell indicates the z/VM ACCOUNT value, which must be 8 characters or less.

Project
The **Project** name that is associated with the AGC Entry.

IAN
When an Intelligent Active Note (IAN) is defined for the AGC Entry, a note icon appears and a tooltip displays the IAN contents.

Locked
When the AGC entry is locked, a lock icon appears.

Status
The status of the AGC entry, which can be **Enabled** or **Disabled**.

Note: A warning message at the top of the window indicates when AGC is not in use.

When you select an entry from the **Existing AGC Entries** table, it populates the **AGC Entry Details** pane. The following fields are displayed in the **AGC Entry Details**:

Name
The name of the AGC entry.

Description
An optional description for the AGC entry.

Updated By/Created By
Indicates the IBM Wave user or process who created and last modified the AGC entry.

Enabled or Disabled
A check box that indicates whether the status of the AGC entry is **Enabled** or **Disabled**.

Associated Metadata
A table that contains one or more rules that determine how AGC classifies the associated metadata for the **AGC Entry**. The table fields indicate:

- **Type** - The type of metadata that is associated with the entry. For example, a project is a metadata type.

- **Value** - The value of the metadata. For example, the project name.
- **Mode** - The mode of the metadata (Bidirectional, Dynamic, or First Discovery). Only one mode can be associated with each AGC rule.

When you right-click on a row in the **Existing AGC Entries**, a menu that contains the following tasks appears (as shown in Figure 12 on page 39):

- **Create New AGC Entry**.
- **Update AGC Entry**.
- **Enable AGC Entry** or **Disable AGC Entry**.
- **Lock AGC Entry** or **Unlock AGC Entry**.
- **Update IAN**, **Read IAN**, or **Delete IAN**.
- **Delete AGC Entry**.

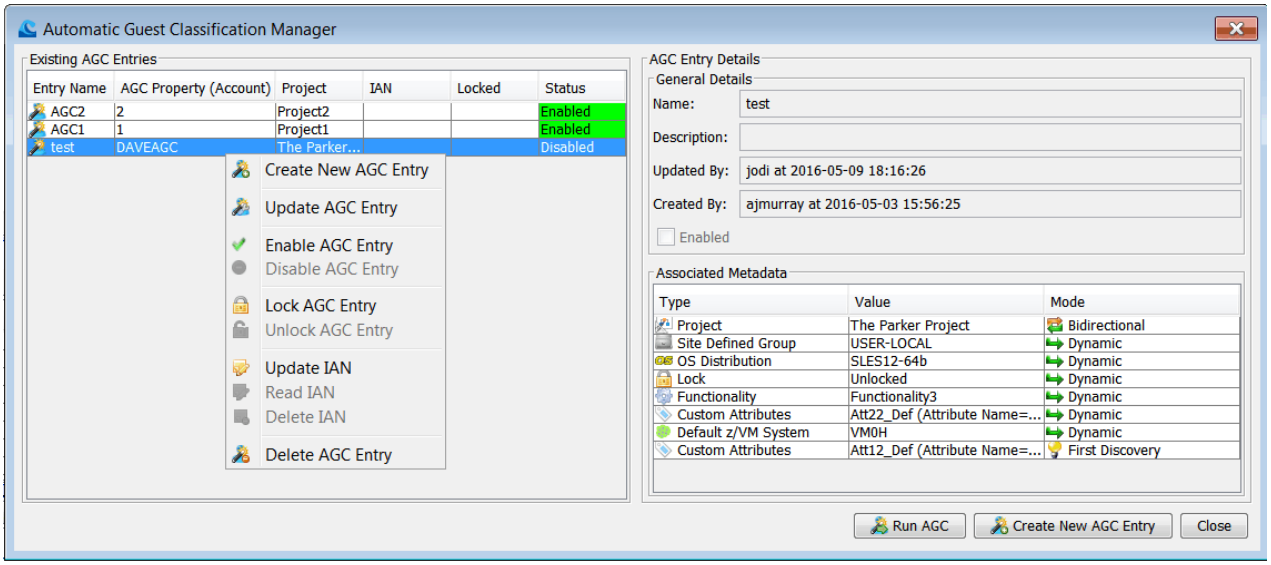


*Figure 12. Actions for **Existing AGC Entries***

Create or Update an AGC Entry
Click **Create New AGC Entry**, or right-click anywhere in the **Existing AGC Entries** pane, and then select **Create New AGC Entry**.

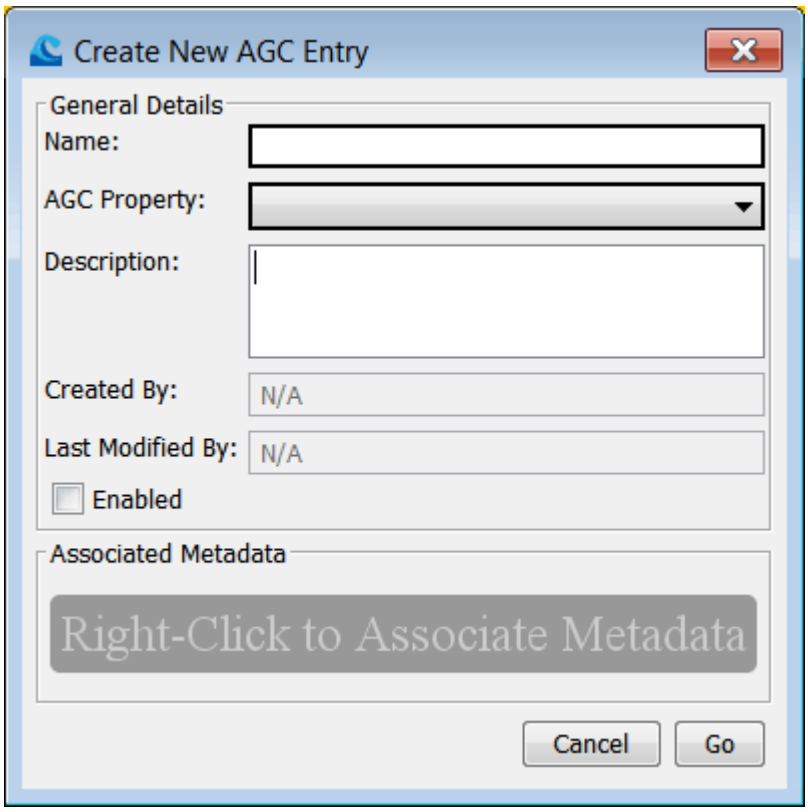


Figure 13. ***Create New AGC Entry pane***

To **Update AGC Entry**, right-click on the row you want to update in the **Existing AGC Entries** pane. The following fields in **General Details** and **Associated Metadata** apply:

- **Name** - The required name for the AGC entry.
- **AGC Property** - The required AGC Property (ACCOUNT value) for the AGC Entry. Click the menu to populate the field with the ACCOUNT values that are known to the Wave server.
- **Description** - An optional description for the AGC entry.
- **Created By and Modified By** - The IBM Wave user who created and last modified the AGC Entry with the associated time stamps.
- **Enabled** - Select the check box to indicate that the AGC entry is enabled. When the check box is cleared, the AGC entry is disabled.
- **Associated Metadata** - A table that lists the associated metadata for the AGC entry. An AGC entry must be associated with a **Project**. Adding more metadata is optional. The following metadata types are supported:
  - Project (required)
  - OS Distribution
  - Default z/VM System
  - Lock – The lock can be used to lock or unlock z/VM guests that use AGC.
  - Site Defined Group

    Note: You cannot use AGC to classify z/VM guests into Internal Site Defined Groups.
  - **Functionality**
  - Custom Attributes

Metadata Association for AGC Entries

To associate metadata with an AGC Entry, right-click in the **Associated Metadata** area, and then click **Create New Metadata Association**.

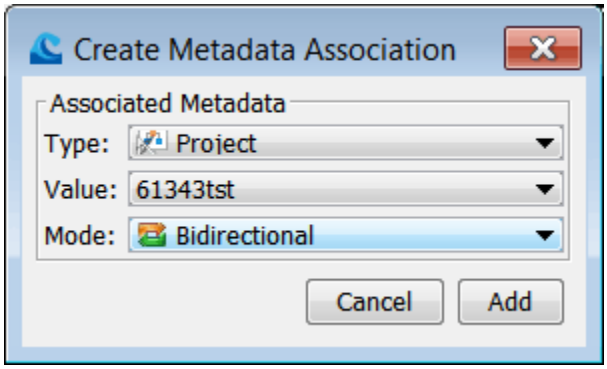


*Figure 14. **Create Metadata Association***

To update an existing metadata association, right-click on an existing entry in the **Associated Metadata** area, and then select **Update** to change the association.

The following fields apply to the **Metadata Association** pane:

- **Type** - The **Type** of metadata to associate. For example, the "Project". Each **AGC Entry** must be associated with a "Project".
- **Value** - The **Value** of the metadata to associate. For example, the "Project" name. The menu is automatically populated when you first select the **Type** or **Value**. For example, when you select "Project" for **Type**, the "**Value**" menu automatically lists all of the "Project" values defined on the Wave server.
- **Mode** - The mode for the **Metadata Association** (Bidirectional, Dynamic, or First Discovery). You can associate only one mode for each metadata association. The "Bidirectional" mode applies only when the **Type** metadata is "Project".

  Important: When AGC is active, you cannot assign a project to a guest if the project does not have the **Bidirectional** mode associated with it. For more information and examples, see "Defining AGC entries" on page 41.

Related tasks

"Running Automatic Guest Classification (AGC)" on page 43
Running Automatic Guest Classification (AGC) associates IBM Wave metadata elements with the z/VM Guest's directory entry.

Related information

"Defining AGC entries" on page 41
"Resolving AGC conflicts and inconsistencies" on page 44

Defining AGC entries

Use the following information to learn how properly define Automatic Guest Classification (AGC) entries.

An administrator can define Automatic Guest Classification (AGC) entries by using the "AGC Manager" on page 37.

AGC entries are composed of the following objects:

AGC Property
Specifies the value in the z/VM guest's directory entry that determines the classification of the IBM Wave metadata. Because only the ACCOUNT value is supported, the value is the primary account value of the z/VM Guest.

Descriptive fields
Specifies a description, any intelligent active note (IAN), and other descriptive data.

Associated metadata

One or more *rules* that specify a set of IBM Wave metadata elements and values that are assigned to the z/VM Guest based on the AGC property value. Each rule is defined with one of the following modes:

Important: When AGC is active, you cannot assign a project to a guest unless the project has the **Bidirectional** mode associated with it.

- **Bidirectional** - The **Bidirectional** mode indicates that if a change is identified in the value of the AGC property, the associated metadata is reassigned. Also, any change to the IBM Wave metadata triggers a change to the value of the AGC property. Currently, only the **Project** metadata can be defined as bidirectional, which means that it is not possible to change other metadata to trigger a change to the AGC property. For example, consider the following AGC entries exist:

Table 5. AGC Property and project metadata

AGC Property (Account)	Associated metadata
ACCNT1	Project=PROJ1, Mode=Bidirectional
ACCNT2	Project=PROJ2, Mode=Bidirectional

When IBM Wave identifies a z/VM Guest with **ACCNT1** as its primary account value, it automatically assigns the guest to **PROJ1**.

Additionally, if you use either of the following methods:

– Use the **Assign Project** action on the z/VM Guest and assign it to project **PROJ1**.

– Drag the z/VM Guest manually into **PROJ1**.

IBM Wave automatically changes the z/VM Guest's primary account value to **ACCNT1**. Also, when the **Update z/VM Guest Aspect** periodic task identifies that the primary account for the guest is changed to **ACCNT1**, IBM Wave automatically changes the project to **PROJ1**.

- **Dynamic** - The **Dynamic** mode indicates that changes identified in the value of the AGC property trigger a change to the associated IBM Wave metadata. As opposed to the **Bidirectional** mode option, changes in the metadata **do not** trigger a change to the value of the AGC property. For example, consider the following AGC entries exist:

Table 6. AGC Property and OS Distribution metadata

AGC Property (Account)	Associated metadata
ACCNT12	Project=PROJ12, Mode=Bidirectional
	OS Distribution=SLES12, Mode=Dynamic
ACCNT14	Project=PROJ14, Mode=Bidirectional
	OS Distribution=SLES12, Mode=Dynamic

When you use the **Assign Account** action on a guest and set the primary account value to **ACCNT12**, IBM Wave automatically changes the OS Distribution of the guest to **SLES12**. Also, when the **Update z/VM Guest Aspect** periodic task identifies that the primary account for the guest is changed to **ACCNT12**, IBM Wave automatically changes the OS Distribution to **SLES12**. However, changing a z/VM Guest's OS distribution to **SLES12** does not change the guest's primary account value.

- **First Discovery** - The **First Discovery** mode is similar to the Dynamic mode, but with one important difference. With First Discovery, the classification happens only when a z/VM guest is discovered by the **Update z/VM Guest Aspect** periodic task, or when the guest is created through IBM Wave actions. Any subsequent change to either the AGC property value or the metadata has no effect on AGC. When you **Run AGC**, you can also apply the metadata associations that use the **First Discovery** mode. To do so, select the **Include First Discovery Metadata Associations** check box as shown in Figure 15 on page 43.

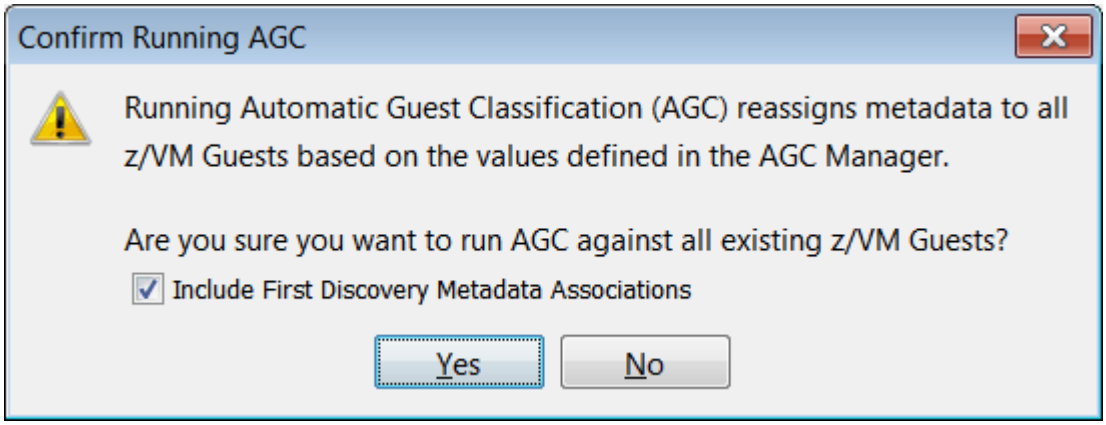


Figure 15. ***Include First Discovery Metadata Associations.***

AGC property values are unique when you define AGC entries. For example, it is not possible to define two AGC entries with the AGC property value **ACCNT1**. However, the metadata that is associated with the AGC entry is not unique. For example, in Table 5 on page 42 you can define the project as **PROJ1** for both **ACCNT1** and **ACCNT2**. The concept creates a one-to-many relationship between the associated metadata and AGC property values. For example, one project can be associated with many primary account values.

Related tasks

“Running Automatic Guest Classification (AGC)” on page 43
Running Automatic Guest Classification (AGC) associates IBM Wave metadata elements with the z/VM Guest's directory entry.

Related information

“Resolving AGC conflicts and inconsistencies” on page 44

Running Automatic Guest Classification (AGC)

Running Automatic Guest Classification (AGC) associates IBM Wave metadata elements with the z/VM Guest's directory entry.

Before you begin

Make sure that you understand the behavior for AGC described in “Automatic Guest Classification” on page 37 and “AGC Manager” on page 37.

About this task

Complete the following steps to run AGC on all discovered z/VM Guests, or use the AGC multiple task action. Running AGC associates the metadata rules with the z/VM Guest's directory entry.

Note: When you run **Auto-Detect**, AGC runs on all newly discovered guests. If you auto-detect a system that already has a directory entry on the Wave server (for example, you auto-detect a system that was previously auto-detected), the most effective method is to run AGC again. New AGC entry classifications are not applied to the system. However, when you run AGC again, if any account values changed outside of IBM Wave, the guest's metadata is classified.

Procedure

1. Understand how to define AGC entries and the metadata for your installation by reviewing the information in “Defining AGC entries” on page 41.
2. Go to **Administrative** > **Site Management** > **AGC Manager**.

   If you need help with any of the fields, review “AGC Manager” on page 37.
3. To enable Automatic Guest Classification (AGC) to run in IBM Wave, go to **Administrative** > **Manage Parameters** and click on the **Functionality** tab. Check the **Use Automatic Guest Classification** check box.

4. On the bottom of the **Automatic Guest Classification Manager**, click **Run AGC**.
   - When you click **Run AGC** in the **Automatic Guest Classification Manager**, it runs AGC on all z/VM Guests on the Wave server.
   - You can also run AGC against one or multiple z/VM Guests by using the multiple task action. Right-click on one or more z/VM Guests, and then select **More Actions** > **Run AGC**.

     Note: The multiple task action runs AGC on only the selected guests.

Results

AGC is running and IBM Wave strictly enforces the AGC rules. For more information, see "Resolving AGC conflicts and inconsistencies" on page 44.

What to do next

Remember: When AGC is active, you cannot assign a project to a guest unless the project associates the **Bidirectional** mode with it.

Resolving AGC conflicts and inconsistencies

The following information explains how you can resolve Automatic Guest Classification (AGC) conflicts and inconsistencies.

For example, in the Table 7 on page 44 scenario, it is not valid for a z/VM Guest to belong to **PROJ1** and not have **ACCNT1** as its primary account value. The opposite is also true. It is not valid for a guest to have **ACCNT1** as its primary account value, but not belong to **PROJ1**. Any failure, conflict, or inconsistency that is detected during AGC processing triggers a special AGC inconsistency status for the z/VM Guest.

Table 7. ***AGC properties and Associated metadata***

AGC Property (Account)	Associated metadata
ACCNT1	Project=PROJ1, Mode=Bidirectional
ACCNT2	Project=PROJ2, Mode=Bidirectional

Under certain circumstances, AGC processing might encounter conflicts or inconsistencies that can be resolved as described in the following sections.

Changing bidirectional metadata

Consider the AGC definitions in Table 8 on page 44.

Table 8. AGC definition conflicts generated by changing bidirectional metadata

AGC Property Value (Account)	Associated Metadata
ACCNT1	**Project=PROJ1**, Mode=Bidirectional
ACCNT2	**Project=PROJ1**, Mode=Bidirectional

Next, consider when an IBM Wave user uses the **Assign Project** action on a z/VM Guest and assigns the guest to the project **PROJ1**. Because "Bidirectional mode" is used in the AGC entries, IBM Wave encounters a conflict during AGC processing; two AGC property values match **PROJ1**. In these scenarios, you are presented with a multiple choice menu to select the correct AGC property value.

AGC inconsistencies or failures

Using the AGC entries that are defined in Table 8 on page 44, consider the following scenarios:

- A z/VM Guest is defined with the primary account **ACCNT3** (or any account value that is not associated to an AGC entry). When the "Update z/VM Guest Aspect" periodic task discovers the guest, AGC processing fails because no AGC entry exists for the AGC property value **ACCNT3**. In this scenario, the z/VM Guest's AGC status indicates an AGC entry mismatch.

- A z/VM Guest is defined with no ACCOUNT value in its directory entry, and no ACCOUNT value in the z/VM Profile specified in the INCLUDE statement. In short, the z/VM Guest is not assigned with any account value. When the "Update z/VM Guest Aspect" periodic task discovers the guest, AGC processing fails because the AGC property is not found in the z/VM guest's definition. In this scenario, the z/VM Guest's AGC status reflects that the guest's AGC property is missing.
- A z/VM Guest is defined as USER with either SYSAFFIN statements or an IDENTITY is defined with several SUBCONFIG statements. Each SYSAFFIN section or SUBCONFIG defines a different account value. When the "Update z/VM Guest Aspect" periodic task discovers the guest, AGC processing fails because the guest is assigned to more than one AGC property value. In this scenario, the z/VM Guest's AGC status indicates a conflict during AGC processing.

IBM Wave Linux shell script repository

Using IBM Wave you can create, edit, and save Linux shell scripts. The scripts can be run on your servers by using the **Execute Script** function. The **Execute Script** function is a multiple task action, and therefore can be executed on one or many virtual servers.

For more information about the **Execute Script** multiple task action, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_execute_script1.html#execscr1.

The user-written scripts are run by the "script executor," which has built in functionality for accepting parameters, debug level, variable reference, and more. The script executor also sets up some initial parameters that you can reference from within the executed script. For a list of parameters, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_appd.html.

IBM Wave includes a script management sub-system that facilitates the creation, deletion, editing, and saving of scripts. For more information, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_the_script_manager.html#thescrmg.

NFS server usage

IBM Wave, upon installation, configures a Network File System (NFS) server on the WAVESRV server to use to run scripts. The scripts that are created or edited through the IBM Wave Script Manager are saved in a directory on the WAVESRV server. Part of the execution process is to mount the script directory by using NFS.

A large Linux installation can span LPARs, CECs, and multiple data centers. Because of network performance issues, security policies, firewalls, or physical limitations, not all guests are able to mount the NFS script directory from the centralized WAVESRV server. IBM Wave provides am alternative for defining more NFS servers to be used in such cases. These NFS servers can be any Linux or Linux Server running a standard NFS Server. Any action that modifies the scripts through the IBM Wave Script Manager automatically syncs the scripts to any defined NFS Server when it is saved in the WAVESRV script directory. In cases where this sync fails (for example if the NFS Server was down when a new script was created), you can manually sync the script in the IBM Wave Script Manager.

When you use the **Execute Script** action, you can manually select the NFS Server from which the script is mounted. You can also define a default NFS Server per z/VM System, so any **Execute Script** action run against guests that are running on that system automatically uses that NFS Server.

For more information about defining more NFS Servers and script syncing, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_update_details1.html and https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_the_script_manager.html#thescrmg.

Note: The sync process runs in the BTS (on the WAVESRV Server) and uses Secure FTP (SFTP[4]) to copy over the scripts from the WAVESRV to any defined NFS Server. Any defined NFS Server must have an SFTP server that is configured, running, and communicating with the WAVESRV server. [5]

[4] SFTP is a standard part of the SSH Server implementation for Linux.
[5] SFTP is a standard part of the SSH Server implementation for Linux.

IBM Wave Linux media repository

IBM Wave contains a repository of Linux installation media (ISO image files). The repository does not contain the actual ISO images, but rather descriptive fields for each ISO image such as the name, description, the Linux distribution, version, service pack, and other fields.

Each Linux media repository points to a network location containing the contents of an ISO image for the installation media of a specific Linux distribution. For a complete list of currently supported Linux distributions, see Appendix A, “Linux distribution support,” on page 165.

The Linux media repositories can be used in conjunction with the Bare Metal Installation (BMI) feature of IBM Wave. For more information, see “Bare-metal installation (BMI) support” on page 25.

The Linux media repository manager can be used to manage these repositories. For more information, see “Adding, updating, and deleting Linux media repositories” on page 96.

Logging

IBM Wave provides several types of logs:

- Regular Log
- GUI Debug Log
- BTS Debug Log
- COR Log Entries

The Regular Log resides in the IBM Wave Database. The GUI Debug Log resides on the client station, under the temporary directory. The BTS Debug Log resides on the WAVESRV server, under /var/log/WAVE. The GUI Debug Log is created only if the "Trace GUI" option is selected from the "User Tasks" menu. The amount of messages in the debug logs (GUI and BTS) is controlled by the Debug Level parameter which can be changed in the WAVE Parameters window. COR Log entries are generated per BTS Request and log the process of that specific request. These can be viewed from the BTS Workunit viewer directly from the GUI.

All actions done by IBM Wave Users and the BTS which change some aspect of the system are logged in the Regular Log. This Log can be viewed and filtered from the GUI Client by selecting the "View Log" option from the "User Tasks" menu.

IBM Wave Users with the Administrator User Type can view all of the log entries, while IBM Wave Users with the Regular User Type can only view their own log entries.

Apart from the traditional logging, for each object, IBM Wave keeps a record of the creating and last modifying IBM Wave User. These fields can be viewed in the property viewer when selecting an object, or (if relevant) from the display panel of that object. These fields cannot be manually changed - IBM Wave updates them whenever necessary.

IBM Wave log messages have the following structure: *WAVxxxyyyc*

Where:

xxx
A code specifying the component which wrote the log entry.

yyy
A code specifying the reason for the log entry.

c
A code specifying the severity of the log entry and can be one of the following:

Informational message (I)
This type of message is issued for events which have completed normally.

Warning message (W)
This type of message is issued for events which have completed with warnings.

Error message (E)
This type of message is issued for events which have completed in error.

Severe message (S)
This type of message is issued for events which have caused the component to fail.

SYSLOGD message routing
You can configure IBM Wave to route messages that are written to the IBM Wave regular log to a specific SYSLOGD.

When changing the log configuration, all messages are also routed to the specified SYSLOGD host. Use the IBM Wave parameters to configure the SYSLOGD host address, and the SYSLOGD facility to which that messages are routed.

From the **IBM Wave main menu** click **Administrative** > **Manage Parameters** > **Audit Log parameters**.

Routing messages to a SYSLOGD host requires the following control statements in the `syslog-ng.conf` file on the SYSLOGD host:

- A source statement like the following example:

```
source s_all {
        # message generated by Syslog-NG
        internal();
        # standard Linux log source (this is the default place for the
        syslog()
        # function to send logs to)
        unix-stream("/dev/log");
        # messages from the kernel
        #file("/proc/kmsg" log_prefix("kernel: "));
        # use the following line if you want to receive remote UDP
        logging
        messages
        # (this is equivalent to the "-r" syslogd flag)
        udp(ip("0.0.0.0") port(514));
};
```

- A filter statement like the following example:

```
filter f_WAVE       { facility(local1); };
```

  Note: In the example above, the Linux SYSLOGD facility uses `local1`. If a different facility is configured, replace "`local1`" with the appropriate facility.

- A destination statement like the following example:

```
destination df_WAVE { file ("/var/log/IBM Wave.log"); };
```

- A log statement like the following example:

```
log { source(s_all); filter(f_WAVE); destination(df_WAVE); };
```

The messages that are routed to the SYSLOGD host must have the following format:

```
DD/MM/YYYY <Originating IBM Wave User>
<Originating IBM Wave Component><Message Code>
<Message Severity><Message>
```

For more information, see “Audit Log parameters” on page 129.

The Attention Required mechanism

IBM Wave provides the **Attention Required** mechanism as a central view to monitor and view entities that are in an error or a warning state. The statuses include inconsistent objects, z/VM Guests that are not connectable, z/VM prototypes that are dissociated, and others.

When the BTS detects a change to a managed object, an Attention Required computation is determined for the object. When the attention required processing yields an attention required entry, it gets added to the IBM Wave database and is updated in the GUI clients.

Attention Required entries are computed per-object, such that if a specific object triggers more than one Attention Required events, the events are grouped under the same Attention Required entry.

For example, a z/VM Guest is not connectable, and the Init for IBM Wave process did not run against the guest yet. The Attention Required entry for that z/VM Guest contains both events. The Attention Required severity is calculated as the highest severity of all the events that are grouped in the Attention Required entry. If the severity for "not connectable" is defined as 70, and the severity for "`Not Inited for IBM Wave`" is defined as 30, the severity for the Attention Required entry is 70.

In the IBM Wave Parameters, you can customize the default severity levels for events, or specify that certain events be ignored. You can also assign a user-defined severity for an Attention Required entry. To do so, go to **Administrative** > **Manage Parameters** and click on the **Attention Required** tab.

The **Attention Required** entries can be viewed in two locations:

1. In the **General Status Viewer**, the **Attention Required** tab contains a table of all of the attention required objects for all of the managed z/VM System LPARs. You can sort the list by using any column header.
2. In the **Current System Viewer**, the **System Status** tab contains the **Attention Required** table. The table contains only the objects that belong to the currently selected z/VM System.

Both tables show only objects that are in the user's scope. Objects that are not in the user's scope are not shown.

Double-clicking an entry in the **Attention Required** table switches the view to the appropriate diagram view and highlights the object in that view. You can right-click the object, and take corrective actions to resolve the error or warning condition.

Note: When an object is not a part of the currently selected z/VM System, a warning message explains that you must manually switch to the system that requires attention.

User defined severity

You can assign a custom severity for an **Attention Required** entry, and filter against user severity values. For example, you might want to filter the views so only the attention required entries with a severity of 30 and above are visible. By default, the user-defined severity for an **Attention Required** entry is equal to the original severity. It is possible to revert the severity to the original severity.

Ignoring Attention Required entries

You can mark an "Attention Required" entry as ignored. Marking the entry as ignored, removes the "Attention Required" indicator from the default views. To view an ignored entry, check the appropriate filter settings in the viewer.

For example, as shown in Figure 16 on page 49, view actions or objects that are ignored, you can check "Ignore" in the **Current System View > System Status** tab or the **General Status Viewer > Attention Required** tab.

When an "Attention Required" entry reverts from the ignored status, its severity reverts to the default or user-defined severity.

Notes:

1. Ignoring an "Attention Required" entry is a global change. It is marked as ignored for all IBM Wave users. It is not possible to ignore an "Attention Required" entry on a user basis.
2. Ignoring an "Attention Required" entry is not persistent. When an "Attention Required" entry is ignored, and the situation is resolved and then recurs, the new "Attention Required" entry is not ignored by default. For example, if a z/VM Guest lost IP connectivity, an "Attention Required" entry is generated and displayed. If the entry was marked as ignored, and then the z/VM Guest regains connectivity and then loses it, a new "Attention Required" entry is generated that is not ignored.
3. When you ignore an entry, you also remove the visual indication about the status of the object. A tooltip remains active to help you understand the "ignored" condition when you hover over an object. For example, if a guest is partially initialized ("inited") and its "Attention Required" entry is ignored,

you cannot recognize the status by looking at the IBM Wave interface. Instead, to understand the status you must either:

- Hover over the item to see the tooltip reminder.
- Review the ignored entries in **Administrative > Manage Parameters > Attention Required Definitions** tab.
- Review the **Current System View > System Status** tab.
- Review the **General Status Viewer > Attention Required** tab.

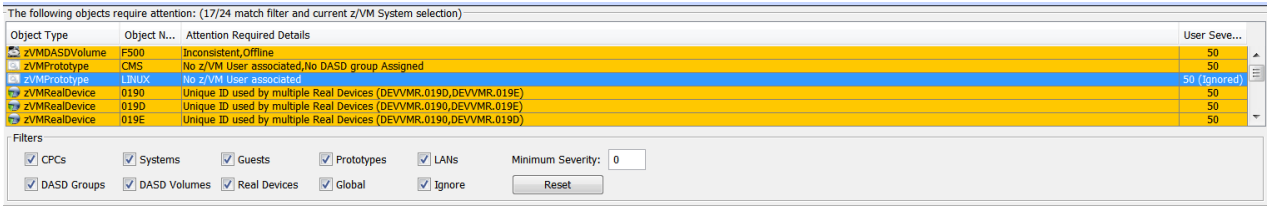


Figure 16. Current System: Ignore filter checked

IBM Wave users

Each IBM Wave user is represented in the IBM Wave database. User definitions can be predefined. Or, when using LDAP Authentication, users can be automatically generated during the first use of IBM Wave for z/VM (after successful authentication).

For complete information, see the following topics:

- Chapter 7, “User management,” on page 147.
- “IBM Wave user authentication” on page 143.
- “Enterprise Directory parameters” on page 127.

IBM Wave user exits

IBM Wave for z/M provides the following type of exits to customize and receive control before or after certain actions occur.

Linux managed z/VM guest exits
Linux executable exits that are stored on managed z/VM guests.

Linux Wave server exits
Linux executable exits that are stored on the Wave server.

REXX exits
The REXX exits, which are written in REXX, must be stored in the 399 minidisk on the WAVEWRKS service machine.

For the IBM Wave exits, see Appendix O, “IBM Wave user exits,” on page 249.

Chapter 2. Installing and customizing IBM Wave

The following information provides step-by-step instructions for installing and customizing IBM Wave for z/VM.

Use the following guide for the complete installation and customization of IBM Wave for z/VM.

- Review "Installation prerequisites" on page 51 to ensure that the prerequisite software is installed and configured for z/VM, Linux guests, the IBM Wave Linux server (WAVESRV), and the workstations that run IBM Wave for z/VM.

  Important: IBM Wave for z/VM supports only American English (AMENG).
- Complete the tasks in "Configuring TCP/IP, SMAPI, and DirMaint" on page 53:
  - "Review the TCP/IP settings" on page 53
  - "Configure SMAPI" on page 54
  - "Authorize DirMaint" on page 55
- Optionally, if you use IBM Wave with Performance Toolkit for z/VM, review "Setting up Performance Toolkit for z/VM" on page 56.
- Configure the security for the IBM Wave environment:
  - If you use IBM Security Server: Resource Access Control Facility (RACF), review "Configuring IBM Wave service machines " on page 57.
  - If you use VM: Secure, see Appendix G, "Configuring VM: Secure," on page 181.
  - If you have no security product, see "Authorizing the service machines in other ways" on page 62.
- Review the following topics for any considerations that apply to your environment:
  - If you are using a z/VM shared directory, review Appendix E, "Shared directory considerations for service machines," on page 173.
  - If you define the service machines within an SSI cluster, review Appendix F, "Considerations for the service machines when working with SSI," on page 177.
- When you are ready to install the Linux server and begin working with IBM Wave, review "Installing IBM Wave for z/VM" on page 62, and then complete the following tasks:
  - "Install the Wave Linux server (WAVESRV)" on page 62
  - "Review the parameter files" on page 30
  - "Start IBM Wave for z/VM" on page 66
- After IBM Wave is installed and configured, periodically review and apply the appropriate service updates, which are made available on IBM Fix Central.

Installation prerequisites

To successfully install IBM Wave for z/VM, you must install and configure the following prerequisite software for the z/VM and Linux systems:

- "Prerequisites for z/VM" on page 51
- "Prerequisites for Linux guests" on page 52
- "Prerequisites for the IBM Wave Linux server" on page 53
- "Prerequisites for workstations that run IBM Wave" on page 53

Note: IBM Wave requires American English (AMENG).

Prerequisites for z/VM

The following prerequisites must be met for each z/VM instance that is managed by IBM Wave:

- IBM Wave requires American English (AMENG).
- If the z/VM instance is to be managed with a secure SSL/TLS connection, the z/VM instance must support TLS 1.0 or later, and must support AES cipher suites.
- The z/VM release must be supported by IBM.

  Note: When you are applying IBM Wave fix packs, be sure to check the readme file for any new z/VM APARs.
- The Linux distribution (also called managed guests) must be listed in Appendix A, "Linux distribution support," on page 165.
- Directory Maintenance Facility for z/VM (DirMaint) or an equivalent directory management product must be installed and active.
- If a security management product is used, it must be RACF or an equivalent security product.
- Storage must be DASD (ECKD) or FCP when an emulated device (EDEV) is being used.
- Networking must be one of the following types:
  - VSwitch
  - Guest LAN (internal z/VM)
  - OSA (ODS) (shared across LPARs).
- TCP/IP must be configured and active with communications protocol of IPV4. (The TCP/IP stack must also be reachable from the client workstations. Telnet, FTP or FTPS, and the z/VM SMAPI ports must be opened to the z/VM managed systems.) For more information about the ports, see "Port reference information" on page 77.
- Telnet (3270) access must be enabled.
- FTP Server must be up and running.
- The SMAPI server that is used by IBM Wave requires authentication by using an existing z/VM guest and its z/VM password. The z/VM guest needs no special privilege classes and might also be called the Authorized API User. See "Configure SMAPI" on page 54 for more information.
- VDISK space of 144000 blocks is needed for use by the Long Service Machine (WAVEWRKL). Make sure the VDISK space allowed for a user by a define command and allowed for the total system is greater than or equal to 144000 blocks. This is usually the default on z/VM systems. The VDISK space can be checked using the z/VM CP commands Q VDISK USERLIM and Q VDISK SYSLIM. For more information about the Q VDISK commands, see one of the following:

  QUERY VDISK (z/VM 6.4)

  QUERY VDISK (z/VM 7.1)

  QUERY VDISK (z/VM 7.2)

Prerequisites for Linux guests

For Linux distributions supported, see Appendix A, "Linux distribution support," on page 165.

For every Linux virtual guest that is managed by IBM Wave, make sure the following packages are installed:

- Binary utilities (binutils)
- SSH Server
- VMCP

The following are required when you use the accompanying feature:

- To run scripts, managed guests require a Network File System (NFS) client.
- For Red Hat Enterprise Linux 7 (RHEL 7), the net-tools package is required.
- For SUSE Linux Enterprise Server (SLES), a CMS file system is required.
- For Ubuntu, a vlan package is required to add a trunk network connection.

Prerequisites for the IBM Wave Linux server

The following prerequisites apply to each Linux system that is running on the IBM Wave server (WAVESRV):

- The Linux distribution on the Wave server must be one of the following:
  - Red Hat Enterprise Linux 7 (RHEL 7) with the net-tools package and other dependency packages.
  - SUSE Linux Enterprise Server 12 (SLES 12) with dependency packages.
- IBM Java 1.8 is required.
- The **deltarpm** package is required.
- The command-line interface (CLI) and the installation of the WebSphere® Application Server Liberty require software to unpack `.tar` and `.zip` files.
- TCP/IP must be configured and active. If a Linux security manager is enabled on the Linux server, verify that it permits you to install IBM Wave.
- If you are using Directory Access, it must be Microsoft Active Directory or LDAP.
- If Directory Access is used with a secure SSL/TLS connection, the directory server must support:
  - TLS 1.0 (or later)
  - AES cipher suites.

Prerequisites for workstations that run IBM Wave

The following prerequisites are for each workstation that is running the IBM Wave client:

- The operating system for the IBM Wave user interface can be Microsoft Windows 7 or Windows 10.
- The IBM Wave client can use Oracle Java 1.8 or IBM Java 1.8.

  Note: For workstations that run the 64-bit version of Windows, 64-bit Java is recommended.
- Browser support must be Microsoft Internet Explorer 9 or Firefox Extended Support Release (ESR) 17.
- TCP/IP must be configured and active for all managed z/VM systems and to the Linux server (WAVESRV).

Configuring TCP/IP, SMAPI, and DirMaint

After the “Installation prerequisites” on page 51 are installed, configure TCP/IP, the z/VM System Management API (SMAPI), and DirMaint. Use the following topics:

- “Review the TCP/IP settings” on page 53
- “Configure SMAPI” on page 54
- “Authorize DirMaint” on page 55

When you are done, review “Setting up Performance Toolkit for z/VM” on page 56.

Review the TCP/IP settings

Review the TCP/IP reference information for the specific z/VM release that is running in your environment. See TCP/IP for VM Publications.

1. Verify that the WAVEWRKS and WAVEWRKL service machines are present and authorized to issue OBEYFILE, and that the port information is usable in the profile TCPIP file.
   a. If not, log on to z/VM by using an authorized ID, and then go to the OBEY section of the profile TCPIP file to add the service machines (as shown in Figure 17 on page 54).

```
; - PROFILE TCPIP created by DTCIPWIZ EXEC on 2 Dec 2015
; - Configuration program run by MAINT at 15:28:06
;   %%File Origin Indicator - DO NOT REMOVE OR ALTER the next
    line%%
;   %%TCPIP%%PROFILE%%STCPIP%%

ASSORTEDPARMS
PROXYARP
ENDASSORTEDPARMS
;
OBEY
OPERATOR  TCPMAINT MAINT MPROUTE REXECD SNMPD
SNMPQE LDAPSRV
WAVEWRKS WAVEWRKL
ENDOBEY
```

Figure 17. TCP/IP: Authorize the service machines

b. While you are in the profile TCPIP file, make sure the port information is present and is not commented out. For example:

```
PORT
20     TCP FTPSERVE   NOAUTOLOG ; FTP Server
21     TCP FTPSERVE  ; FTP Server
23     TCP INTCLIENT ;  Telnet Server
```

Figure 18. TCP/IP: Check the port information

- If you use Performance Toolkit for z/VM, you can update the profile TCPIP file for it now. See "Setting up Performance Toolkit for z/VM" on page 56.

2. Enable the z/VM FTP server by following the steps for the specific z/VM release that your installation is running. For the correct TCP/IP and z/VM AUTOLOG information, see the information for your version of z/VM in TCP/IP for VM Publications.

   - Check NETSTAT. If FTPSERVE is not present, enter:

     ```
     XAUTOLOG FTPSERVE
     ```

   - Make sure the FTPSERVE can start automatically after a TCP/IP restart. Check the AUTOLOG section of the profile TCPIP file for more information.

TCP/IP planning information

TCP/IP Planning and Customization (z/VM 6.4)

TCP/IP Planning and Customization (z/VM 7.1)

TCP/IP Planning and Customization (z/VM 7.2)

Configure SMAPI

Before you install IBM Wave, you must configure the z/VM Systems Management Application Programming Interface (SMAPI).

IBM Wave uses SMAPI to perform various tasks on z/VM. Before IBM Wave is installed, SMAPI must be configured and the IBM Wave authorized API user ID must have permission to issue all API requests and must be defined as a normal (neither LBYONLY nor AUTOONLY) user.

Note: The configuration steps for SMAPI can differ based on the version and release of z/VM in your installation. See "SMAPI Installation and Configuration: Setting up and Configuring the Server Environment" on page 55 for more information. Use the information that matches the z/VM release that your installation is running.

Important: If an IBM Wave action fails with a return code of -3, verify that you have enough SMAPI worker machines to handle the workload. If needed, add more SMAPI worker machines.

After changing SMAPI authorizations, restart SMAPI by entering the following **FORCE** and **XAUTOLOG** commands:

```
FORCE VSMGUARD
XAUTOLOG VSMGUARD
```

SMAPI Installation and Configuration: Setting up and Configuring the Server Environment

- Authorizing API Requests (z/VM 6.4)
- Authorizing API Requests (z/VM 7.1)
- Authorizing API Requests (z/VM 7.2)

SMAPI APARs

For a list of SMAPI APARs, see:

z/VM System Management

Note: SMAPI APARs are often *not* included with RSUs.

Authorize DirMaint

The following steps are required to authorize DirMaint as your directory manager.

IBM Wave for z/VM includes service machines that must be authorized to use DirMaint facilities. Use the following steps to authorize DirMaint access.

Before you begin, review the following topics:

- If you are using a z/VM shared directory, go to Appendix E, “Shared directory considerations for service machines,” on page 173.
- If you define the service machines within an SSI cluster, see Appendix F, “Considerations for the service machines when working with SSI,” on page 177.
- If you are using **VM: Secure**, review Appendix G, “Configuring VM: Secure,” on page 181.

1. Modify CONFIG DATADVH. When you configured DirMaint for use, you created a configuration override file to change the IBM supplied CONFIG DATADVH file. This file typically has two extra characters that are appended to the file name (such as CONFIG*xx* DATADVH). Use the **DIRM CMS LIST CONFIG* DATADVH *** command to retrieve the list, which is applied in reverse alphabetical order. In the following commands, the extra characters are represented by *xx*.

   ```
   DIRM SEND CONFIGxx DATADVH
   ```

2. Make sure the ALLOW_ASUSER_NO_PASS entries in the DirMaint configuration files are adequate for the version of z/VM that you are running.
   Add the following statements to the CONFIG*xx* DATADVH file:

   ```
   ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
   ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
   ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
   ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
   ALLOW_ASUSER_NOPASS_FROM= WAVEWRKS *
   ALLOW_ASUSER_NOPASS_FROM= WAVEWRKL *
   ALLOW_ASUSER_NOPASS_FROM= WAVEWRKC *
   ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
   ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
   ```

   You must also add ALLOW_ASUSER_NO_PASS lines for any other API worker service machine.

3. IBM Wave for z/VM requires that the DATAMOVE function is available. If you did not activate DATAMOVE when you originally configured DirMaint for use, do so now. Add the following statements:

```
DISK_CLEANUP= YES
DATAMOVE_MACHINE= DATAMOVE * *
```

File the changes to save them. After you save the changes to the CONFIG*xx* DATADVH file, send it back to DirMaint.

```
DIRM FILE CONFIGxx DATADVH
```

4. Get a copy of the AUTHFOR CONTROL file from DirMaint.

```
DIRM SEND AUTHFOR CONTROL
```

Add the following statements to the AUTHFOR CONTROL file:

```
ALL VSMGUARD * 140A ADGHMOPS
ALL VSMGUARD * 150A ADGHMOPS
ALL VSMWORK1 * 140A ADGHMOPS
ALL VSMWORK1 * 150A ADGHMOPS
ALL VSMWORK2 * 140A ADGHMOPS
ALL VSMWORK2 * 150A ADGHMOPS
ALL VSMWORK3 * 140A ADGHMOPS
ALL VSMWORK3 * 150A ADGHMOPS
ALL WAVEWRKS * 140A ADGHMOPS
ALL WAVEWRKS * 150A ADGHMOPS
ALL WAVEWRKC * 140A ADGHMOPS
ALL WAVEWRKC * 150A ADGHMOPS
ALL WAVEWRKL * 140A ADGHMOPS
ALL WAVEWRKL * 150A ADGHMOPS
```

You must also authorize any other API worker service machine.

Save the changed AUTHFOR CONTROL file and send it back to DirMaint:

```
DIRM FILE AUTHFOR CONTROL
DIRM RLDD
DIRM RLDC
```

Run the following command to ensure that the AUTHFOR CONTROL file was successfully updated:

```
DIRM FOR ALL AUTHFOR ?
```

5. For IBM Wave to Auto-Detect a z/VM system and to install its service machines, you must define at least one DirMaint DASD group and add at least one DASD volume to DirMaint.

When DASD regions are defined in the z/VM system, the region names must match the names of the DASD volumes on which they are stored.

Note: IBM Wave does not support multiple regions per DASD volume.

DirMaint APARs

For a list of DirMaint APARs, see:

IBM: DirMaint APARs

Setting up Performance Toolkit for z/VM

This topic describes how to set up IBM Wave to work with Performance Toolkit for z/VM.

The following information is specific to using IBM Wave with Performance Toolkit for z/VM.

For IBM Wave to work correctly, the short service machine, WAVEWRKS, must be authorized to retrieve information from the Performance Toolkit for z/VM virtual machine. Add the following to the FCONRMT AUTHORIZ file in the z/VM machine that is running Performance Toolkit for z/VM:

```
<SYSTEM-NAME><Short Service Machine Name>DATA CMD
```

The configuration steps listed here are done when you set up Performance Toolkit for system use. It is important to review and make sure the necessary files are available for IBM Wave.

- PROFILE EXEC
- FCONRMT PROFILE
- PERFSVM PROFILE EXEC
- The PROFILE EXEC for the PERFSVM user ID enables monitor sampling for the samples and events that you need. All monitors must be active. One method is to add the lines to the PROFILE EXEC file of the Performance Toolkit for z/VM virtual machine:

```
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'CP MONITOR SAMPLE ENABLE STORAGE'
'CP MONITOR SAMPLE ENABLE USER ALL'
```

- The virtual machine communication facility (VMCF) interface for Performance Toolkit must be activated. To do so, add the FC MONCOLL VMCF ON statement to the FCONX $PROFILE file.

For the complete steps for setting up Performance Toolkit, review the publications that match the z/VM release that is running in your environment:

- **z/VM 6.4**
  - z/VM: Performance Toolkit Guide
  - z/VM: Performance Toolkit Reference
- **z/VM 7.1**
  - z/VM: Performance Toolkit Guide
  - z/VM: Performance Toolkit Reference
- **z/VM 7.2**
  - z/VM: Performance Toolkit Guide
  - z/VM: Performance Toolkit Reference

Configuring IBM Wave service machines

The following steps are required before you attempt to Auto-Detect and manage a z/VM system with RACF active. The service machines must exist before you can enter the RACF commands to authorize the service machines. The following information explains how to create the service machines and the ID by using the **DIRM ADD** command in DIRMAINT file, which must be done from the MAINT user ID.

- If you do not use RACF, go to the following topic and select the appropriate path: Chapter 2, “Installing and customizing IBM Wave,” on page 51.
- If you use a z/VM shared directory configuration, review Appendix E, “Shared directory considerations for service machines,” on page 173.
- If you define the service machines within an SSI cluster, review Appendix F, “Considerations for the service machines when working with SSI,” on page 177.

Creating the service machines

Important:

- IBM Wave supports only American English (AMENG), which displays as OPTION -LANG-AMENG in the service machines.

- The minidisk passwords must be the same for the service machine 191, 399, and Linux media repository disks.

1. If you are running a single system z/VM, create the `WAVEWRKS DIRECT` file and copy the following text into it:

   Where *PASSWORD* is the password for all three of the service machines.

```
USER WAVEWRKS PASSWORD 128M 512M ABCDEFG
IPL CMS
MACHINE ESA 4
OPTION LNKNOPAS DIAG88
CONSOLE 0009 3215
SPOOL 000C 3505 A
SPOOL 000D 3525 A
SPOOL 000E 1403 A
*CSLTAG01: WAVE-INTERNAL
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK TCPIP 0592 0592 RR
LINK MAINT 0193 0193 RR
```

2. Now create the z/VM user:

```
DIRM ADD WAVEWRKS
```

3. Create the minidisks:

```
DIRM FOR WAVEWRKS AMD 191 3390 AUTOG 100 TEST
WR PW READ WRITE MULTI

DIRM FOR WAVEWRKS AMD 399 3390 AUTOG 100 TEST
WR PW READ WRITE MULTI
```

   The minidisk passwords must be the same for the service machine 191, 399, and Linux media repository disks.

   - Where *TEST* is the name of the storage group to be used.
   - Where *READ* is a read password that is chosen for the minidisk.
   - Where *WRITE* is a write password that is chosen for the minidisk.
   - Where *MULTI* is a multi-password that is chosen for the minidisk.

4. Create the `WAVEWRKC DIRECT` file and copy the following information into it:

```
USER WAVEWRKC <PASSWORD> 128M 512M ABCDEFG
IPL CMS
MACHINE ESA 4
OPTION LNKNOPAS LNKE DIAG88
CONSOLE 0009 3215 T
SPOOL 000C 3505 A
SPOOL 000D 3525 A
SPOOL 000E 1403 A
*CSLTAG01: WAVE-INTERNAL
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK TCPIP 0592 0592 RR
LINK MAINT 0193 0193 RR
LINK WAVEWRKS 0191 0191 RR
LINK WAVEWRKS 0399 0399 RR
```

   Note: On the `CONSOLE 0009 3215 T` statement, you can set up another user ID to observe the console output from WAVEWRKC, WAVEWRKL, or both. Change the CONSOLE statement to "CONSOLE 0009 3215 T *userid* OBSERVER", where *userid* is a user ID such as OPMGRM1.

5. Create the z/VM user:

```
DIRM ADD WAVEWRKC
```

6. Create the WAVEWRKL DIRECT file and copy the following into it:

```
USER WAVEWRKL PASSWORD 128M 512M ABCDEFG
IPL CMS
MACHINE ESA 4
OPTION LNKNOPAS DIAG88
CONSOLE 0009 3215 T
SPOOL 000C 3505 A
SPOOL 000D 3525 A
SPOOL 000E 1403 A
*CSLTAG01: WAVE-INTERNAL
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK TCPIP 0592 0592 RR
LINK MAINT 0193 0193 RR
LINK WAVEWRKS 0191 0191 RR
LINK WAVEWRKS 0399 0399 RR
```

7. Create the z/VM user for the WAVEWRKL service machine:

```
DIRM ADD WAVEWRKL
```

8. Log on to the MAINT USERID and ensure that the WAVEWRKS, WAVEWRKL, and WAVEWRKC service machines are not logged on (if necessary, **FORCE** logoff).
9. Format the 191 and 399 minidisks.
   a. Link to the 191 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 191 1191 WR"
```

      1) Format the minidisk:

```
"FORMAT 1191 J"
```

      2) When asked for a label, enter:

```
WAV191
```

      3) Detach the disk:

```
"rel J (DET"
```

   b. Link to the 399 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 399 1399 WR"
```

      1) Format the minidisk:

```
"FORMAT 1399 J"
```

      2) When asked for a label, enter:

```
WAV399
```

      3) Detach the disk:

```
"rel J (DET"
```

Authorizing the service machines with RACF

1. Because the service machines are populated by the auto-detect process, you must grant all the necessary RACF permissions before you attempt to auto-detect a system. Set the passwords for the three IBM Wave service machines:

```
RAC ALU WAVEWRKS PASS(PASSWORD) NOEXP
RAC ALU WAVEWRKL PASS(PASSWORD) NOEXP
RAC ALU WAVEWRKC PASS(PASSWORD) NOEXP
```

   where *PASSWORD* is a password, which is chosen by you, for each of the service machines.

2. Give the service machines access to the minidisks.

   a. Grant WAVEWRKC and WAVEWRKL service machines access to the WAVEWRKS 191 and 399 minidisks:

```
RAC PERMIT WAVEWRKS.191 ID(WAVEWRKC WAVEWRKL) CLASS(VMMDISK) ACC(ALTER)

RAC PERMIT WAVEWRKS.399 ID(WAVEWRKC WAVEWRKL) CLASS(VMMDISK) ACC(ALTER)
```

   b. Give the WAVEWRKS service machine access to the DIRMAINT.1DF minidisk:

```
RAC PERMIT DIRMAINT.1DF ID(WAVEWRKS) CLASS(VMMDISK) ACC(READ)
```

   c. Give the WAVEWRKS service machine access to the AUTOLOG1 191 and AUTOLOG2 191 minidisks:

```
RAC PERMIT AUTOLOG1.191 CLASS(VMMDISK) ID(WAVEWRKS) ACCESS(ALTER)

RAC PERMIT AUTOLOG2.191 CLASS(VMMDISK) ID(WAVEWRKS) ACCESS(ALTER)
```

   The IBM Wave short service machine, WAVEWRKS, on each managed z/VM System must be permitted to modify the AUTOLOG1 191 DISK and AUTOLOG2 191 disk, if it exists. This permission allows the WAVEWRKS machine to facilitate the creation or modification of certain z/VM entities that are defined as permanent.

   For more information, see “Configuring AUTOLOG” on page 35.

   d. Give all IDs access to read WAVEWRKS 399 minidisk:

```
RAC PERMIT WAVEWRKS.399 ID(*) CLASS(VMMDISK) ACC(READ)
```

   e. Give the WAVEWRKS service machine access to operate as an alternative ID for FTPSERVE:

```
RAC PERMIT WAVEWRKS CLASS(VMBATCH) ID(FTPSERVE) ACCESS(CONTROL)
```

3. FORCE all of the following IDs that apply to the level of z/VM that your installation is running:

```
WAVEWRKS, WAVEWRKC, WAVEWRKL, VSMREQIN, VSMREQIU, VSMREQI6, VSMEVSRV,
FTPSERVE, PERFSVM
```

4. The WAVEWRKC service machine needs to be authorized to link read/write to target clone minidisks. In RACF, you must grant WAVEWRKC the OPERATIONS privilege by using the following command:

```
RAC ALU WAVEWRKC OPERATIONS
```

5. Shut down DIRMAINT:

```
DIRM SHUTDOWN
```

6. To allow the IDs to validate passwords:
   a. Create a profile called DIAG088 in the VMCMD class with a default access of NONE:

   ```
   RAC RDEFINE VMCMD DIAG088 UACC(NONE)
   ```

   b. For each ID you forced off or shut down in step “3” on page 60, enter the following command (substituting the *USERID* field):

   ```
   RAC PERMIT DIAG088 CLASS(VMCMD) ID(USERID) ACCESS(READ)
   ```

7. After each ID, identified previously, is granted access:
   a. Activate the VMCMD class:

   ```
   RAC SETROPTS CLASSACT(VMCMD)
   ```

   b. **XAUTOLOG** each ID including DIRMAINT. For example:

   ```
   XAUTOLOG DIRMAINT
   ```

8. Start the service machines (WAVEWRKS, WAVEWRKC, and WAVEWRKL).
9. Make sure the service machines are exempt from the **FOR.C** and **FOR.G** commands.

   All Wave service machines must be exempt from the **FOR.C** and **FOR.G** classes to be able to run CP commands on behalf of another ID.

   To make the service machines exempt, enter the following commands:

```
RAC RDEFINE VMXEVENT USERSEL.WAVEWRKS
RAC RDEFINE VMXEVENT USERSEL.WAVEWRKL
RAC RDEFINE VMXEVENT USERSEL.WAVEWRKC

RAC RALTER VMXEVENT USERSEL.WAVEWRKS ADDMEM(FOR.C/NOCTL FOR.G/NOCTL)
RAC RALTER VMXEVENT USERSEL.WAVEWRKL ADDMEM(FOR.C/NOCTL FOR.G/NOCTL)
RAC RALTER VMXEVENT USERSEL.WAVEWRKC ADDMEM(FOR.C/NOCTL FOR.G/NOCTL)

RAC SETEVENT REFRESH USERSEL.WAVEWRKS
RAC SETEVENT REFRESH USERSEL.WAVEWRKL
RAC SETEVENT REFRESH USERSEL.WAVEWRKC
```

   You should consider auditing these LINK requests; the default is NOAUDIT. To enable auditing of the requests, RALTER each VMXEVENT profile shown in this step with the ADDMEM(AUDIT) keyword from a user authorized to control auditing, and then REFRESH the profile. Depending upon your organization's separation of duties with regard to security policies, you might need to have a different person enable auditing.

10. Disks are defined automatically during the bare metal installation (BMI) process, but they need to have RACF Security Server for z/VM permissions defined.

    For example, to define one or more disks by using the default addresses 400 to 40X for each disk:

    ```
    DIRM FOR WAVEWRKS AMD 40x 3390 AUTOG 100 TEST
    WR PW READ WRITE MULTI
    RAC PERMIT WAVEWRKS.40x ID(*) CLASS(VMMDISK) ACC(READ)
    ```

    where:

    TEST is the name of the storage group to be used.
    READ is a read password, that you choose, for the minidisk.
    WRITE is a write password, that you choose, for the minidisk.
    MULTI is a multi-password, that you choose, for the minidisk.

For more information, see the z/VM: RACF Security Server Security Administrator's Guide and the following RACF-related information.

Tailoring the DIRMAINT Service Machine: CONFIG DATADVH
Step 5. Select RACF-Specific Characteristics (z/VM 6.4)
Step 5. Select RACF-Specific Characteristics (z/VM 7.1)
Step 5. Select RACF-Specific Characteristics (z/VM 7.2)
RACF Security Server for z/VM publications
For z/VM 6.4
For z/VM 7.1
For z/VM 7.2

Authorizing the service machines in other ways

If you run without any external security manager product, CP's authorization controls will govern the service machines. The directory entries provided previously (“Creating the service machines” on page 57), along with minidisk passwords, give the service machines sufficient authorization to function.

If you have a non-IBM security product, follow the instructions from the manufacturer. You should expect to control similar resources, and give the service machines equivalent access, but the command syntax will differ.

Installing IBM Wave for z/VM

To install IBM Wave for z/VM, follow these steps:

1. “Install the Wave Linux server (WAVESRV)” on page 62.
2. “Start IBM Wave for z/VM” on page 66.

Installation concepts

The WAVESRV server is a Linux server that runs as a virtual server under z/VM. The server runs a database that contains a repository of all the IBM Wave managed objects, which comprises the full IBM Wave database.

Install the Wave Linux server (WAVESRV)

Before you begin

- You should configure the `/var` file system with a type of XFS and at least 10 GB of free space.

The amount of disk space that will be required will vary based on a number of factors. When determining how much disk space you will need, you must take into consideration the amount of historical data, backups, and logs that you intend to retain, the number of z/VM systems that will be managed by IBM Wave, and any other factors that might influence disk usage.

When Linux is running normally, it uses the `/var` file system to contain data that is written and read. This includes such data as system logs, application logs, spooled data, mail, and the IBM Wave database. Keeping in mind that `/var` is typically the file system that experiences the most rapid growth, you should mount `/var` as its own file system. This is done to prevent the main file system from being filled to capacity by variable data, resulting in system instability.

When installing the IBM Wave server, you should create `/var` with an initial size of 10 GB as an XFS file system logical volume under the Logical Volume Manager (LVM) so that it can be expanded as needed. In addition, this offers a chance to help optimize disk I/O by configuring the disks used by the LVM Volume Group that will contain `/var` to use such options as multipathing and HyperPAV aliases. If you are required to retain log data for long periods of time due to governance or regulations, consider a larger initial size. IBM recommends against using any copy-on-write snapshotting file systems such as btrfs or ZFS for `/var` on the IBM Wave server due to the potential scaling and performance issues associated with data that is highly variable.

For information about log rotation options for total logging size, see "Wave server log options" on page 134.

- You should configure the `/usr` file system with at least 1 GB of free space.

  It is best to have at least 1 GB of free space on the Wave server's `/usr` disk.
- Depending on where you place fix pack files on your Wave server, you might need to make sure `/tmp` has enough space.

  If you use a separate disk for `/tmp` and you download the fix pack and RPM files to `/tmp`, it is best to have at least 300 MB of free space on `/usr` and a bit more on `/tmp`.
- When you install SUSE Linux Enterprise Server (SLES), select **Minimum System**. For an example of the Linux WAVESRV directory entry, see Appendix C, "A sample WAVESRV directory entry," on page 169.
- If the RPM version of IBM Java is not available, IBM Wave also supports the "InstallAnywhere" version.

  See the Java Platform Standard Edition download site:

  http://www.ibm.com/developerworks/java/jdk/linux/download.html
- IBM Wave supports RACF SURROGAT LOGON (or similar function, depending upon your ESM) for the z/VM guest that is running the WAVESRV. For more information, see the planning information for the z/VM version in your environment (at the end of this topic).

Procedure

1. Using the distribution's standard tools, make sure the following packages are installed:
   - For SUSE Linux Enterprise Server 12 (SLES 12):
     - apache-prefork
     - nfs-kernel-server
     - mariadb
     - mariadb-client
   - For RedHat Enterprise Linux 7 (RHEL 7):
     - httpd-2.2.3-11
     - mariadb
     - mariadb-server
     - nfs-utils-1.3.0-0.8

After the setup is complete, and the designated WAVESRV Linux server is started, complete the following steps.

2. FTP the IBM Wave RPM file, `IBM-Wave-1.20-1.s390x.rpm`, from the installation media to a directory on the WAVESRV Linux server.
3. Use an SSH client from your workstation to log in as `root`.
4. IBM Wave allows clients to access its services using a host name or IP address. If you want clients to use a host name, you must make sure that WAVESRV's Linux server is configured with a client-resolvable host name.

   If the host name was not set up by the Linux installation process, configure the host name so that the `hostname -f` command returns a usable value. If you want the IBM Wave client, the REST APIs, or both to be accessible using a host name, the host name value you configure must be resolvable by all client endpoints.

   The host name or IP address you select (*wave_server_ip_address*) must also be listed as a subject alternative name (SAN), properly typed as a host name or IP address in the server's certificate that you will obtain in step "5" on page 64 and then install in step "10" on page 65.
5. Obtain a valid server certificate for *wave_server_ip_address*, signed by a certificate authority that your enterprise's workstations trust.

   See Appendix L, "Managing Wave's server certificate," on page 195.
6. Verify that the TCP/IP ports are available as described in "Port reference information" on page 77.
7. Get the newest IBM Wave for z/VM fix pack from IBM Fix Central.

   Go to:

   IBM Wave on Fix Central

   and follow the instructions in the `readme` file that comes with the fix pack.

   Remember: IBM Wave fix packs are cumulative.
8. Install the RPM.

   Unpack the fix pack that you downloaded to a directory on your WAVESRV Linux server. Run the `doUpdate.sh` script with the following parameters to install the RPM and update to the newest fix pack:

   ```
   doUpdate.sh -i rpm_file_name
   ```

   a. Wave's BTS default TCP/IP *port-number* is 3300 to listen for requests. If your installation blocks port 3300 and you must use another port, add `--btslistenport` *`port-number`* to the end of the `doupdate` invocation above. For example:

      ```
      doUpdate.sh -i rpm_file_name --btslistenport 4305
      ```

      Any valid available port in the range 0-65535 can be specified.

   b. The `doUpdate.sh` script will prompt for the client-resolvable IP address or host name that you chose in step "4" on page 64 (*wave_server_ip_address*). You can choose one of the values displayed or enter one of your own choosing.

   c. The `doUpdate.sh` script will prompt for a keystore password. The value is selected, managed by, and should only be known to you. IBM Wave stores the value in its database, so your enterprise's workstations can validate the Wave server's identifying certificate when creating secure network connections. You can change the value after installation.
9. Create the IBM Wave application administrator's credentials.

   Because you are installing IBM Wave for the first time, the `doUpdate.sh` script will prompt you for an administrative user name. When you respond, the script will create this IBM Wave user and assign it all defined IBM Wave permissions, making this IBM Wave user a superuser (see "Understanding user types and roles" on page 147 for more information). IBM Wave will generate and display the

new user's initial password in the script output. You should provide those credentials to the Wave application administrator, who must change the password at first login. Figure 19 on page 65 shows a sample of some successful script output.

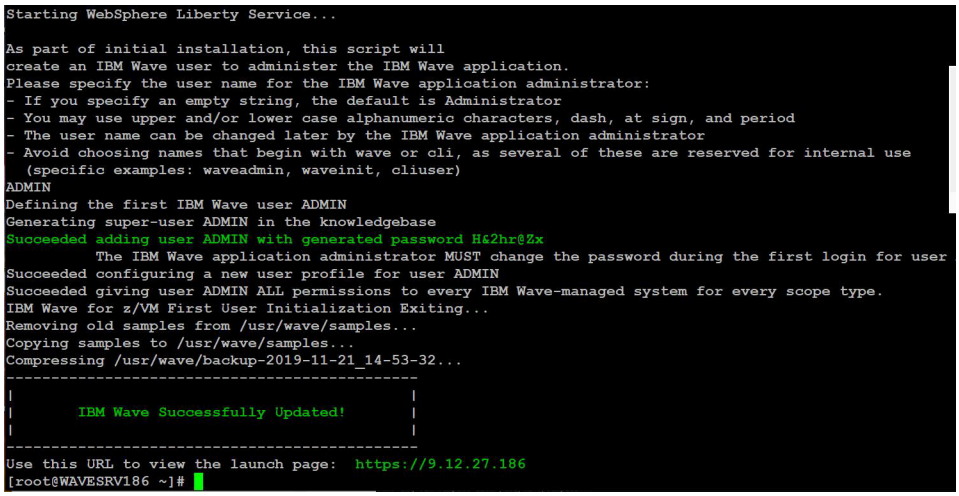


Figure 19. Creating the application administrator's credentials: sample script output

Note: In the event that all earlier steps completed successfully, but an error occurred while creating the administrative user, you can try creating the user again without uninstalling and then re-installing IBM Wave. To do this, use the `doUpdate.sh` script as follows:

```
doUpdate.sh [ -createfirstuser | --createfirstuser ]
```

10. Install the server certificate you obtained in step "5" on page 64 or configure IBM Wave client to skip validation of the server's identity.
    a. IBM recommends installing a valid server certificate, because this allows you to fully secure network connections to the Wave server. See Appendix L, "Managing Wave's server certificate," on page 195.
    b. If you were unable to obtain a valid server certificate, you can manually disable certificate validation checking by following the procedure in "Disabling Wave server certificate validation in the IBM Wave client" on page 142. *This option is less secure; IBM does not recommend using it in production environments.* When you configure Wave this way, Wave's login dialog will display a security warning during every login.

What to do next

After you install the WAVESRV server, give the IBM Wave server URL to the IBM Wave application administrator, along with the credentials you created in step "9" on page 64. The IBM Wave server URL, which you configured previously using the `doUpdate.sh` script, is `https://`*wave_server_ip_address*. The script displays the fully-resolved value at the end of its output whenever it runs successfully; for example, in Figure 19 on page 65, it is `9.12.27.186`.

The IBM Wave application administrator can now "Start IBM Wave for z/VM" on page 66.

Planning information for z/VM and RACF

- z/VM: CP Planning and Administration
- Defining Users
- Tailoring the DIRMAINT Service Machine: CONFIG DATADVH, Step 5. Select RACF-Specific Characteristics
- RACF Security Server for z/VM publications

Start IBM Wave for z/VM

After the Wave server Linux administrator installs the IBM Wave server, the IBM Wave application administrator can start the IBM Wave client using a web browser. To launch IBM Wave client, enter the client-resolvable IP address or host name for the Wave server (provided to you by the Wave server Linux administrator) in your browser's URL bar.

Your *IBM Wave home page* appears in your browser as shown in Figure 20 on page 66. This page is also called the *IBM Wave launch page*.

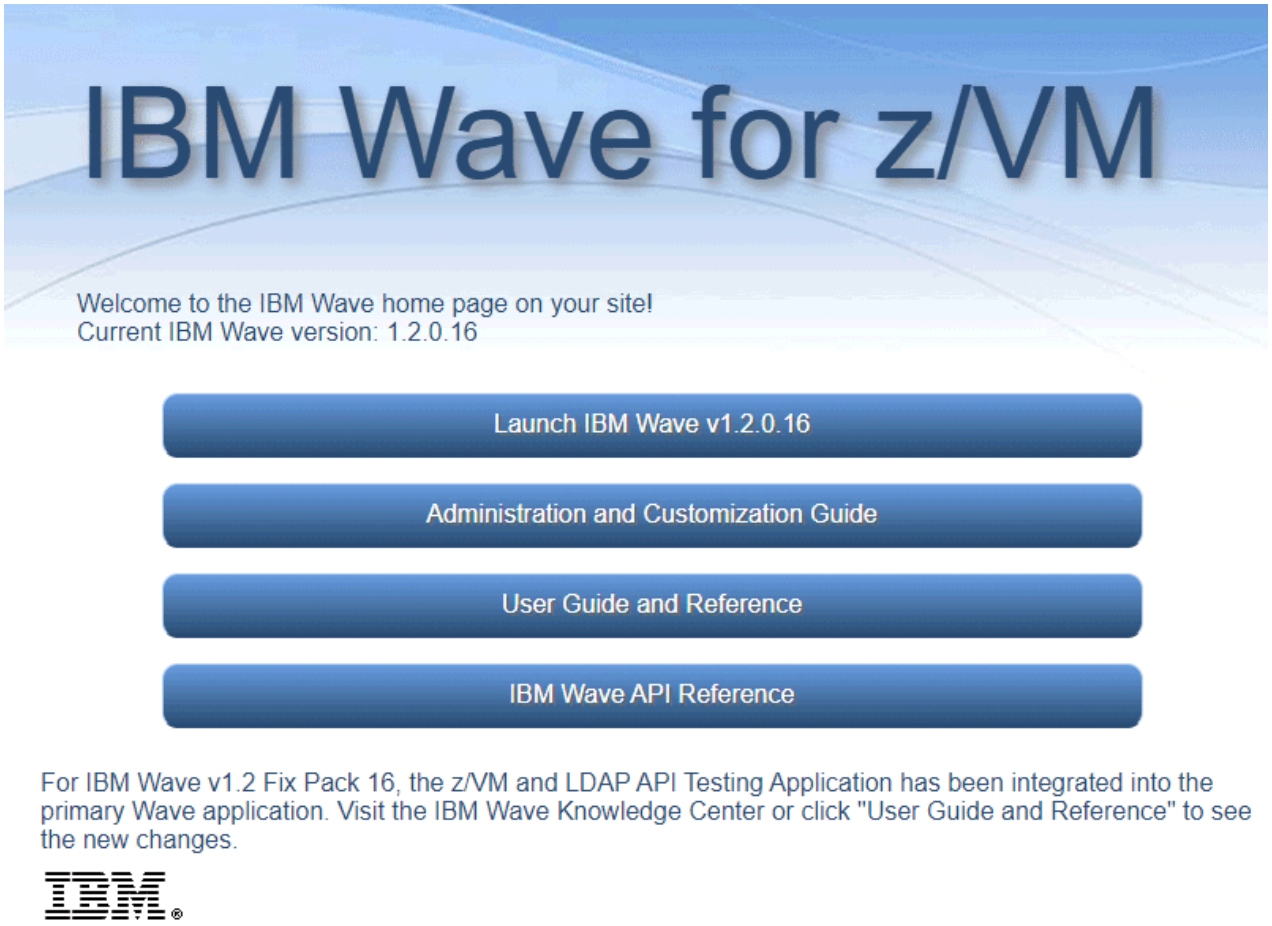


Figure 20. Welcome to your IBM Wave home page

On the IBM Wave home page, you can select the following options.

- Click **Launch IBM Wave** to get the login screen that starts the IBM Wave for z/VM client.
- Click **Administration and Customization Guide** or **User Guide and Reference** to access the information for the current fix pack.
- Click **IBM Wave API Reference** to access the REST API reference documentation.

The IBM Wave client uses Java Web Start. Your home page's first link downloads, installs, and launches the IBM Wave client on your Windows workstation. This process also creates a link to IBM Wave for z/VM on your Windows desktop and in the Windows start menu.

When the application starts, it will prompt for your Wave credentials (as shown in Figure 21 on page 67, assuming your Wave server Linux administrator configured IBM Wave as recommended). To start the IBM Wave client, enter your **User Name** and **Password**, and then click **Log In**. On your very first login, IBM Wave will force you to change your password, so its value is no longer known to your Wave server Linux administrator.

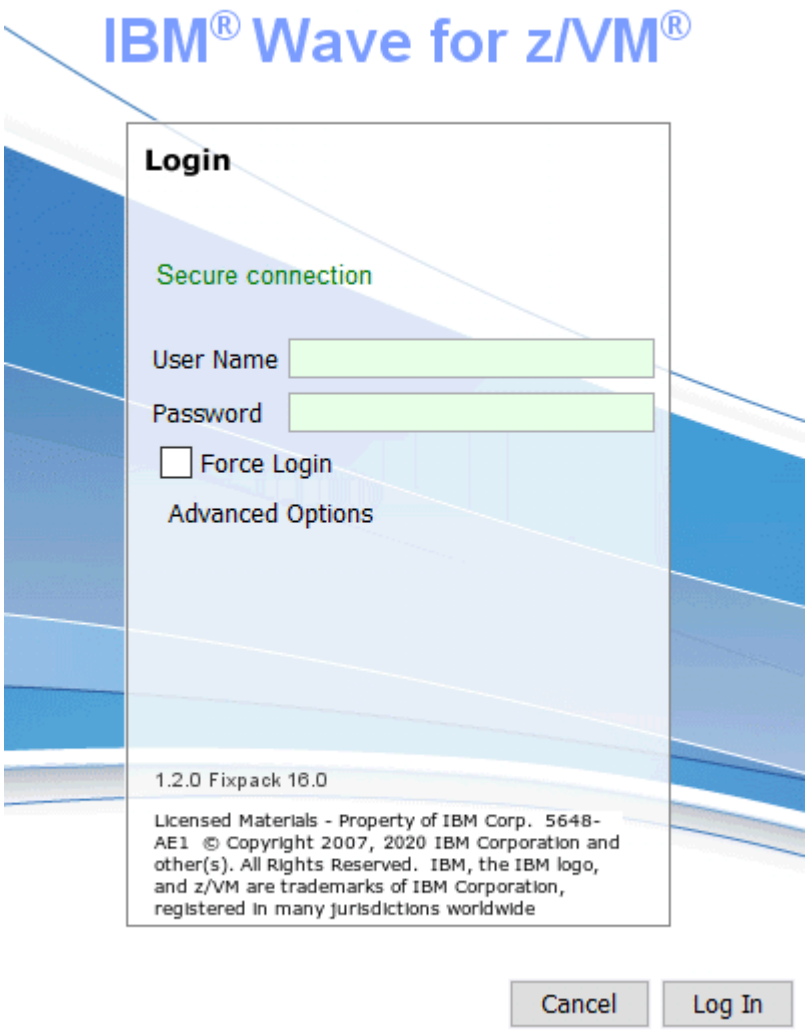


Figure 21. IBM Wave login window (when configured as IBM recommends)

The following options are also available:

- Click **Advanced Options** to run IBM Wave in debug mode.
- Click **Force Login** to start a new session and log off other open session by the same user ID.

If you see a `Failed to connect` pop-up window or a red security warning in the credentials prompt instead of the green `Secure connection` message shown in Figure 21 on page 67, this means your Wave server Linux administrator configured IBM Wave differently; see Diagnosing your connection status for more details.

Next, you should review IBM Wave's parameters, especially those related to security policies, to make sure they align with your enterprise policies. "IBM Wave security tasks" on page 137 will help you locate security-specific parameters. It is also a good idea to review the others during initial installation (see Chapter 5, "System customization," on page 113).

After you have customized your installation, you should use the External Entities Manager ("External Entities Manager" on page 85) to begin adding the IBM Z mainframes, z/VM systems, and storage controllers. Then, you are ready to prepare the z/VM guests for initial use of IBM Wave for z/VM (defining IBM Wave users for the Linux).

- To add a CPC, see "Add New CPC" on page 98 (and Add New CPC).

- To add a z/VM system to a CPC, see "Running the Auto-Detect Wizard" on page 71, "Configuring AUTOLOG" on page 35, and Create New z/VM System.

  Important: Make sure you review the manual steps that you must take to add a z/VM System to IBM Wave management. See "Running the Auto-Detect Wizard" on page 71 and "Configuring AUTOLOG" on page 35.
- To initialize already-existing Linux virtual servers for IBM Wave use, see "Linux Login Security Options" on page 140 and Init Users for IBM Wave use.

  Note: This step must be done before any IBM Wave interaction with the z/VM virtual server (such as cloning, convert to prototype, connect to LAN, and other tasks).
- To give users access to IBM Wave for z/VM, see Chapter 7, "User management," on page 147.

After you complete these steps, IBM Wave is fully operational.

For more information about starting the IBM Wave client after installation, see "Overview of the IBM Wave client" on page 5.

Upgrading IBM Wave for z/VM

Before you begin

- IBM Wave fix packs are cumulative.
- The Wave server Linux administrator performs this task.
- For file system size considerations, see the **Before you begin** topic in "Install the Wave Linux server (WAVESRV)" on page 62.
- Make sure the Wave server Linux administrator has a current backup of the Wave database (see "Backup IBM Wave Database" on page 100).
- Existing fix packs can be re-applied to replace files on the Wave server and the service machines, or to change the BTS listening port used for GUI and CLI requests.

Procedure

1. Get the newest IBM Wave for z/VM fix pack from IBM Fix Central.

   Go to:

   IBM Wave on Fix Central

   and follow the instructions in the `readme` file that comes with the fix pack.
2. Install the fix pack.

   Unpack the fix pack that you downloaded to a directory on your Wave server. Change your working directory to the directory containing the unpacked files, and run the `doupdate.sh` script with no parameters to install the latest cumulative fix pack without making any configuration changes, as shown in the first example below. If you want to make configuration changes while upgrading (or afterward), you can add parameters as shown in the second example below.

   ```
   ./doUpdate.sh
   ```

   a. The `doUpdate.sh` script might prompt for the client-resolvable IP address or host name for the Wave server. You can choose one of the values displayed or enter one of your own choosing.

   ```
   ./doUpdate.sh
   or
   ./doUpdate.sh --btslistenport port-number
   ```

b. If you need Wave's BTS server to bind to a port other than the currently-defined port (the default is 3300), use `--btslistenport` *port-number* to change the currently-defined TCP/IP port number that Wave's BTS server uses to listen for requests.

Testing the connection to SMAPI and the service machines

Follow this procedure to test the connections to SMAPI and to the IBM Wave service machines on an existing (or new) z/VM system.

About this task

Before you start IBM Wave for the first time, use this task to test the connections to the z/VM Systems Management Application Programming Interface (SMAPI) and the IBM Wave service machines for your IBM Wave for z/VM environment.

Important: IBM recommends that you test the connections *before* clicking on **Create** (the **Add a new system** action on a CPC) or **Update** (the **Update details** action on a system) because the action will fail and you will have to re-enter your changes if the connection parameters are incorrect.

Procedure

1. From the Hardware Viewer, right-click an existing z/VM system (LPAR) and click **Update Details**.
   You will see the **Update z/VM System** window:

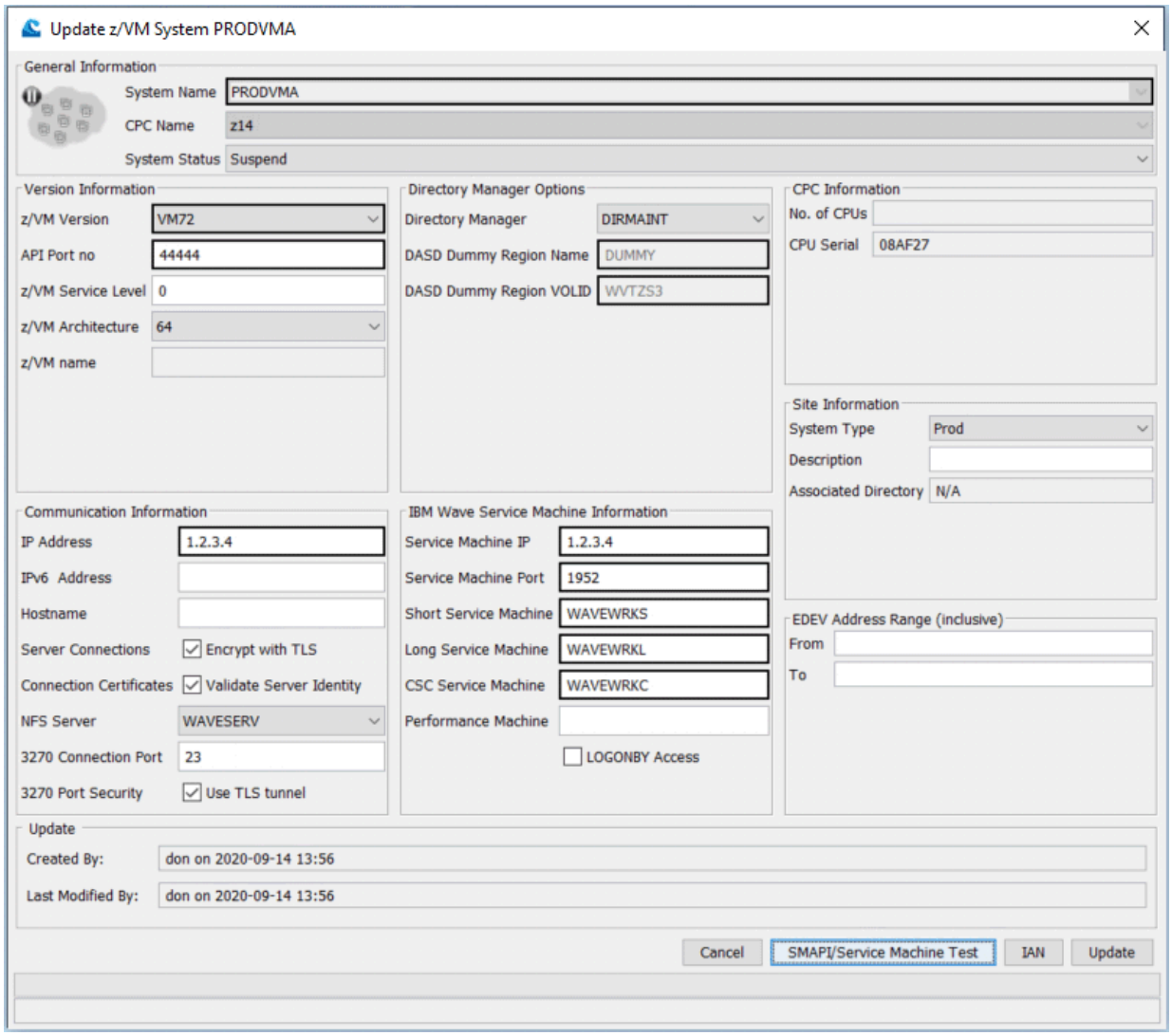


Figure 22. Update z/VM System Window

2. Click on **SMAPI/Service Machine Test**.
   You will see the **z/VM SMAPI and IBM Wave service machine Connection Test** window:

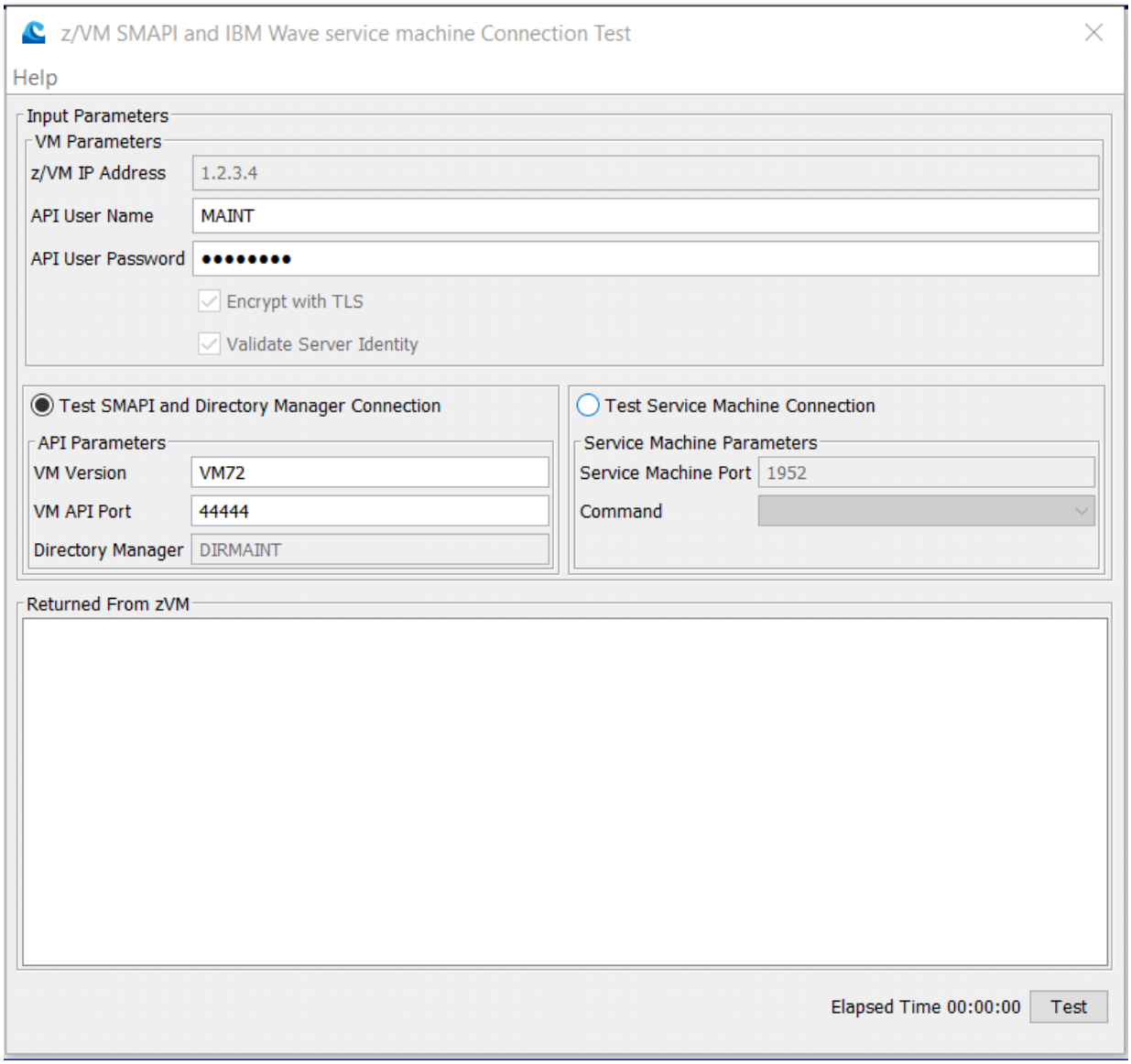


Figure 23. z/VM SMAPI and IBM Wave service machine Connection Test Window

3. Select **Test SMAPI and Directory Manager Connection** or **Test Service Machine Connection**.
4. Fill in the fields with the information for your IBM Wave for z/VM environment.
   This window includes the following fields:
   - **z/VM IP Address** - The IPv4 address for the z/VM system.
   - **API User Name** - The name of the authorized SMAPI user.
   - **API User Password** - The password of the authorized SMAPI user.
   - **VM Version** - The z/VM operating system (OS) version.
   - **VM API Port** - The port on which the API server is listening.
   - **Directory Manager** - The type of directory manager that is installed on the z/VM system.
   - **Service Machine Port** - The port on which the WAVEWRKS service machine listens for incoming requests.
   - **Command** -
     - **NETWORK_NETSTAT** - Use this command to query the network status of the service machine.

- **DASD_REGION_QUERY_DIRM** - Use this command to query the status of real direct access storage devices (DASDs) on the DIRMAINT dummy region.
- **GUEST_QUERY_ID** - Use this command to query the ID of the service machine.

5. Click on the **Test** button.

Results
Depending on the configuration you choose, Wave sends the appropriate queries to test the SMAPI or service machine connection.

Running the Auto-Detect Wizard

The IBM Wave for z/VM Auto-Detect Wizard is an intuitive interface that guides you through the installation of a newly-created z/VM system.

About this task

Use the **Auto-Detect Wizard** to add a z/VM system to IBM Wave management. Select the newly-defined z/VM system, and then from the IBM Wave main menu, click **Auto Detect** > **Run Autodetect Wizard**. The **Auto-Detect Wizard** contains the following steps.

Procedure

1. Review **Step 1 - Welcome** in Figure 24 on page 71, which contains an overview of the **Auto-Detect** steps.

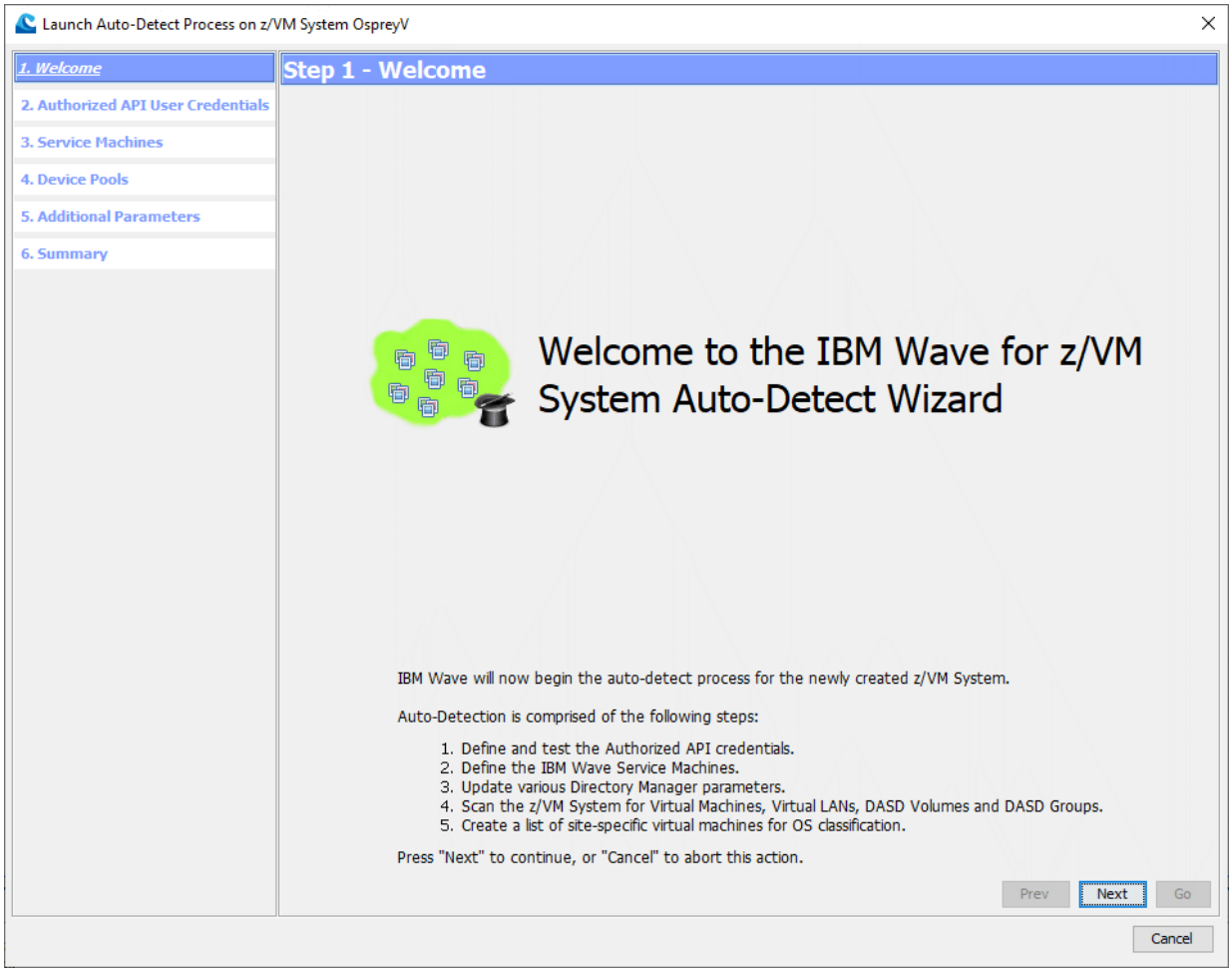


Figure 24. Step 1 - Welcome

2. Enter the authorized API user credentials in **Step 2 - Authorized API User Credentials**, as shown in Figure 25 on page 72.

   The authorized API user credentials must be an existing z/VM user ID. The auto-detect process must connect to the z/VM system's API (SMAPI) server to verify that the API server is up, running, and accessible to IBM Wave. See information about the Authorized API User and SMAPI authentication in "Prerequisites for z/VM" on page 51.

   After you enter the user name and password, click **Next**. Mouse (or hover) over the **Password** field on this panel (as shown in Figure 25 on page 72) to see information about password length and valid characters.

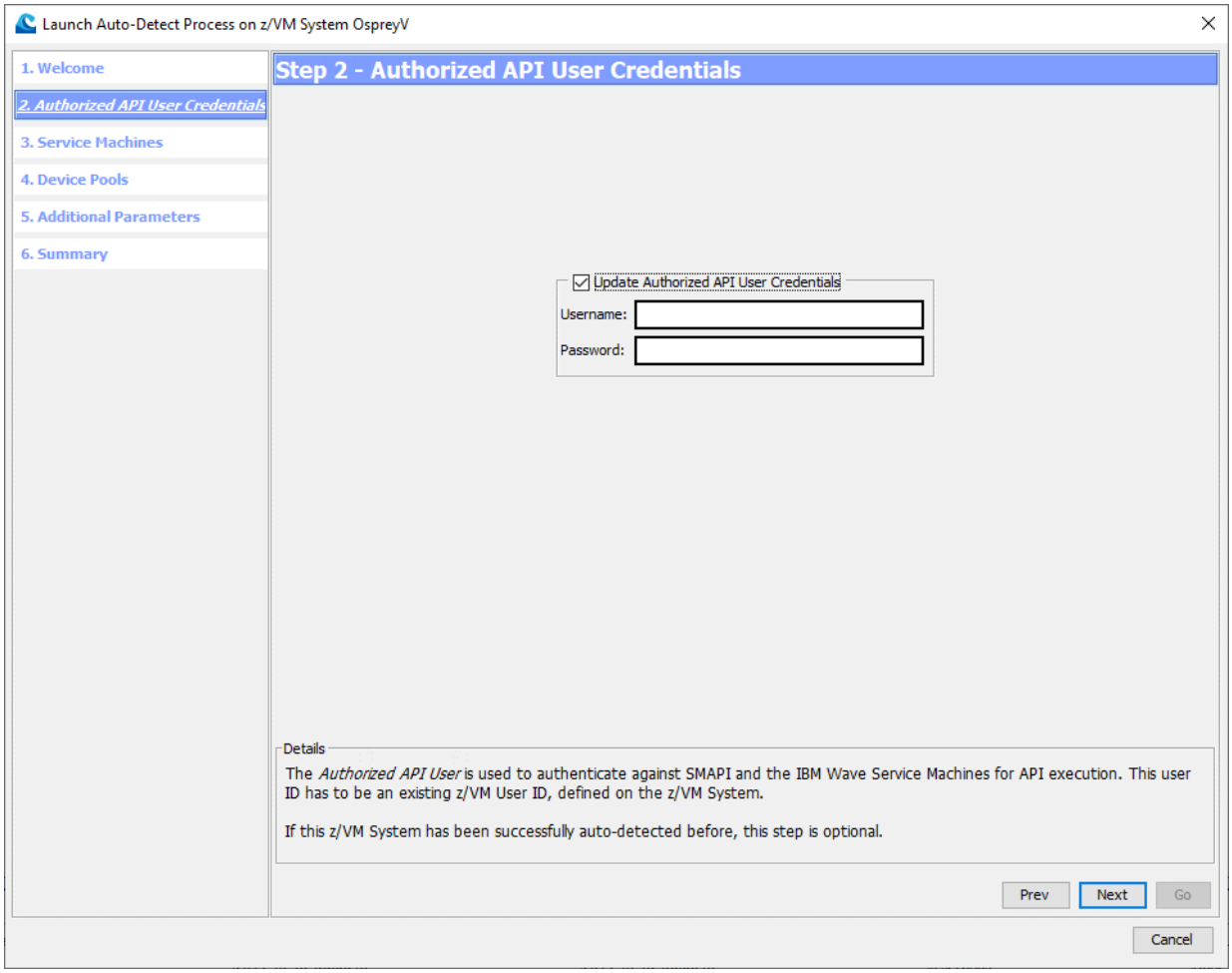


Figure 25. Step 2 - Authorized API User Credentials

3. Define the service machines in **Step 3 - Service Machines** as shown in Figure 26 on page 73.

   The following fields are available in **Step 3 - Service Machines**:

   - **Short Service Machine (WAVEWRKS), Long Service Machine (WAVEWRKL), CSC Service Machine (WAVEWRKC)** - The options for the service machine to create and populate IBM Wave for z/VM. When the service machines are found running a compatible version, the option is **No Action**.
   - **Service Machines Password** - The z/VM password for the service machines. Mouse (or hover) over the **Service Machines Password** field on this panel (as shown in Figure 26 on page 73) to see information about password length and valid characters.
   - **Use DASD Group** - An optional DASD group name on which the service machines are installed.
   - **Use DASD Volume** - An optional DASD volume name on which the service machines are installed. For more information, see "Configuring AUTOLOG" on page 35.

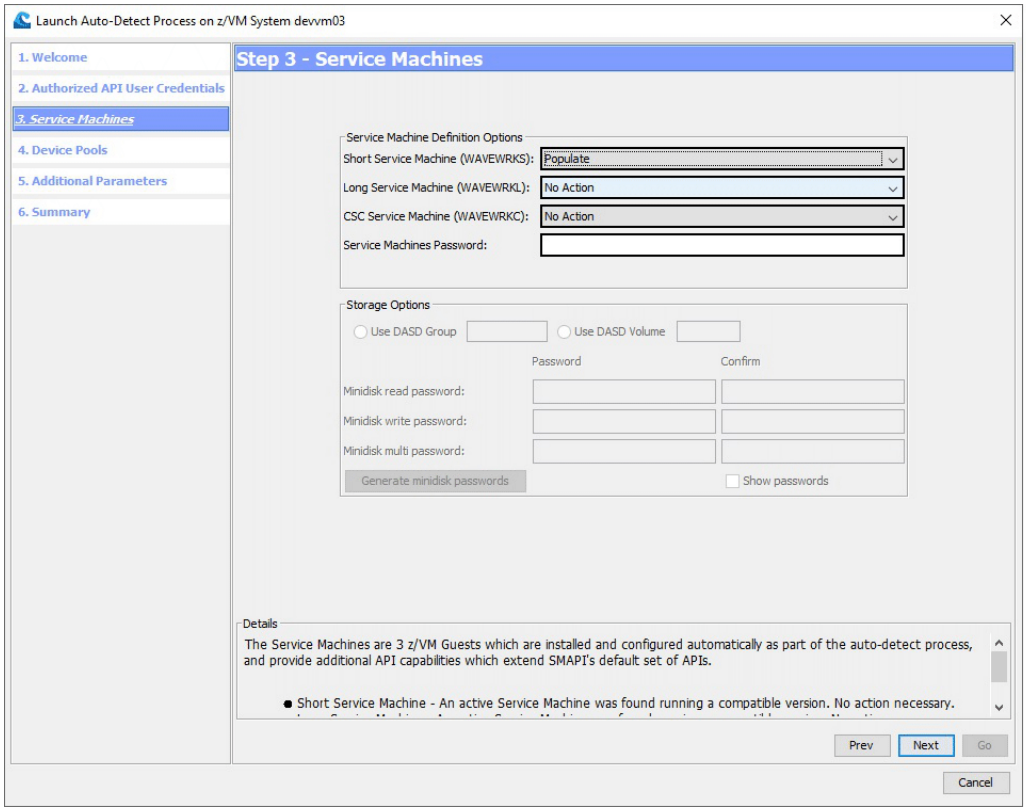


Figure 26. Step 3 - Service Machines

4. Specify the device pools for DASD, OSA, HIPER, and FCP in **Step 4 - Device Pools** as shown in Figure 27 on page 74.

   The DASD, OSA, HIPER, and FCP parameters define the Device Pool to which IBM Wave adds the newly-discovered real devices when it auto-detects the z/VM system. The device pool can either be an existing device pool, or the default-named device pool for the z/VM System.

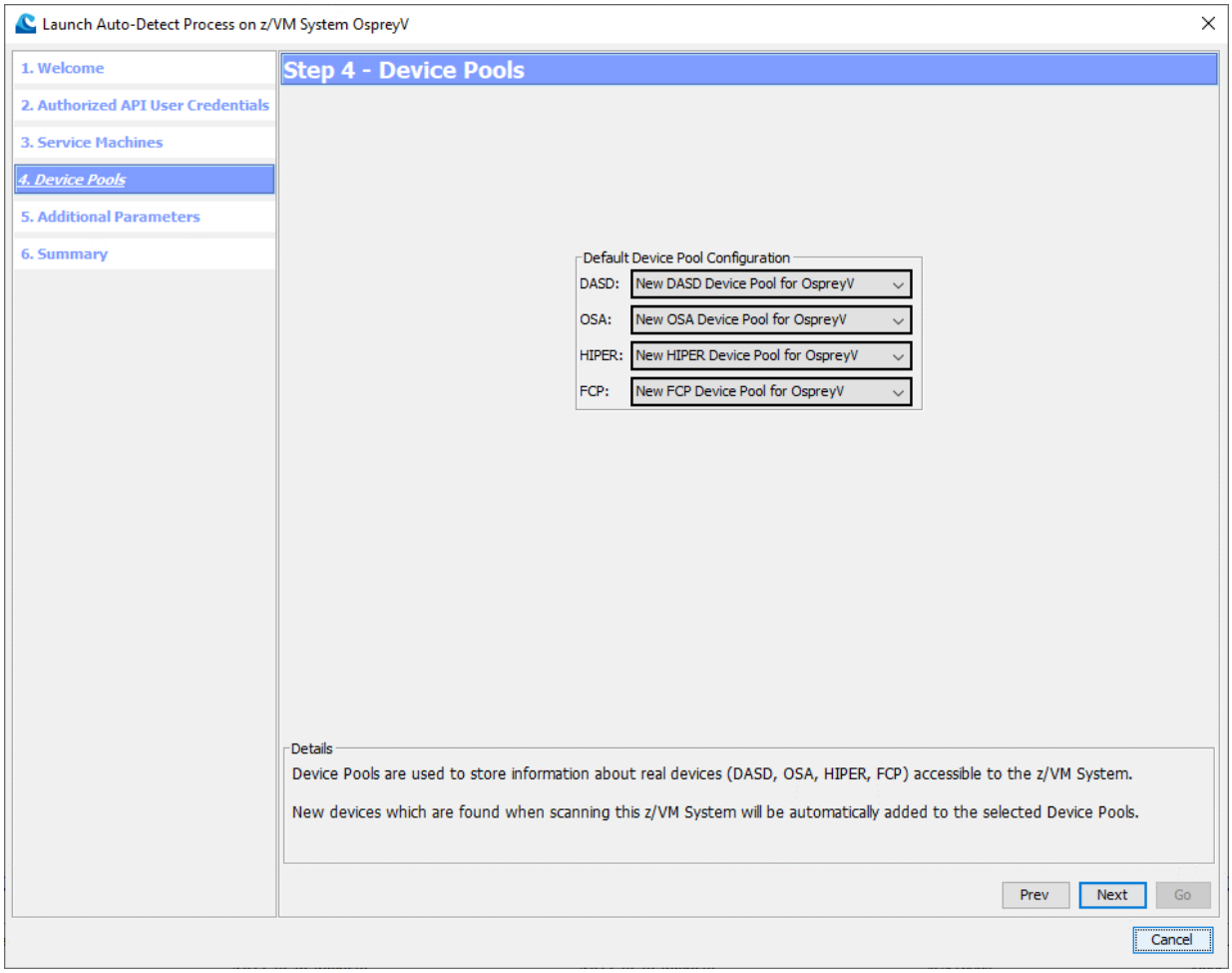


Figure 27. Step 4 - Device Pools

5. Provide the additional configuration for TCP/IP and DASD in **Step 5 - Additional Parameters** as shown in Figure 28 on page 75.

   - **Guest Running the TCP/IP Stack** - The name of the TCP/IP virtual server in the z/VM System.
   - **Minidisk Address for TCP/IP Executables** - The minidisk address with the TCP/IP executable files.
   - **Guest Running the Directory Manager** - The name of the z/VM Guest running the directory manager, which by default is DIRMAINT or "VMANAGER" for VM: Secure or VM: Direct.
   - **Use Autolog Facility Using Guest** - The name of the AUTOLOGx machine. The default is AUTOLOG1. AUTOLOG usage by IBM Wave is optional. It might not be appropriate for an installation with complex logic in the `AUTOLOG PROFILE` file. Before you decide to use the AUTOLOG facility in your installation, read the details about how IBM Wave uses the `PROFILE.EXEC` in “Configuring AUTOLOG” on page 35.
   - **Define Dummy Region** - DIRMAINT does not permit the creation of empty storage groups. To ease the creation of a new storage group without needing to immediately assign a DIRMAINT region to it, Wave creates a "Dummy Region". The "Dummy Region" is one cylinder in size. It is created during **Auto-Detect** phase when you add a z/VM System to IBM Wave management (providing that the directory manager on the z/VM System is DIRMAINT). Although the region must be defined on a real DASD Volume, it is never physically allocated. It is a placeholder for empty storage groups.

     Restriction: The "Dummy Region" must never be defined larger than one cylinder.

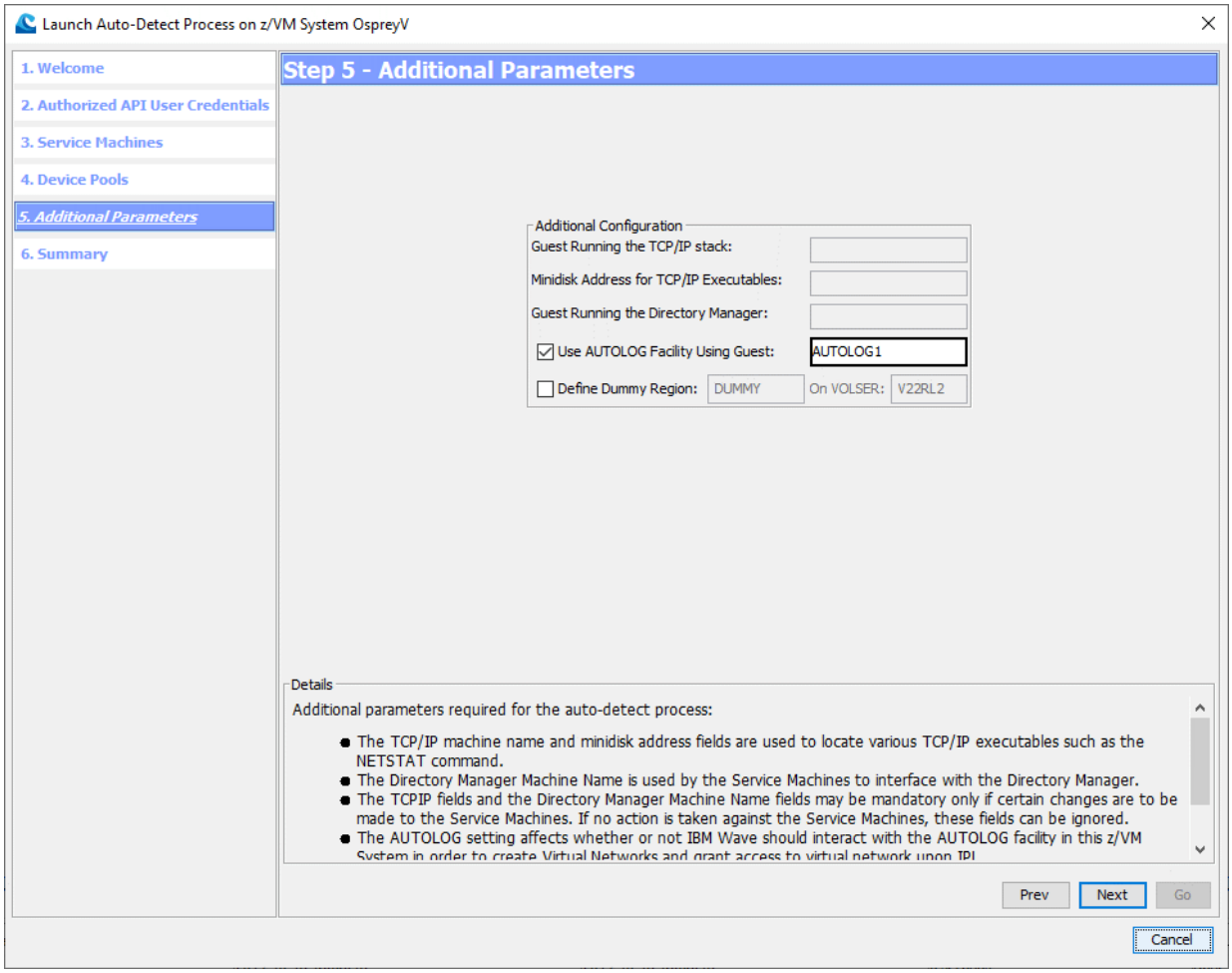


Figure 28. Step 5 - Additional Parameters

6. Review **Step 6 - Summary** as shown in Figure 29 on page 76 for a summary of the configuration options you selected in the previous steps.

   It is important to review the information in “Configuring AUTOLOG” on page 35. After you update the `PROFILE.EXEC`, and are satisfied with the changes, click **Go**.

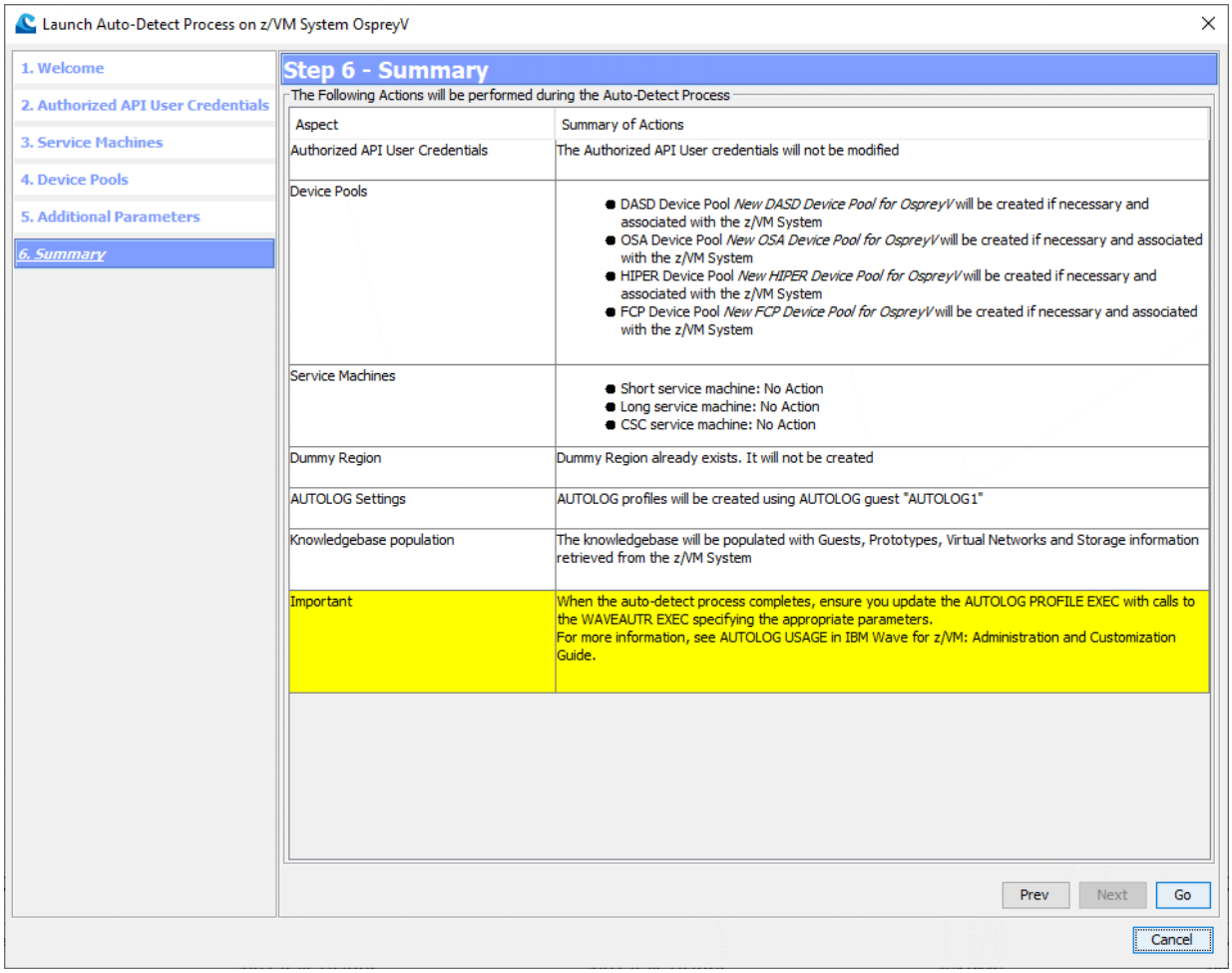


Figure 29. Step 6 - Summary

If necessary, you can click **Previous** to go back and change a previous step.

Results

The new z/VM system is configured to start working in IBM Wave for z/VM.

Related information

“Configuring AUTOLOG” on page 35
“Start IBM Wave for z/VM” on page 66

Port reference information

The following information is provided as a reference to help you easily locate the default port numbers for IBM Wave for z/VM.

Wave server ports

Table 9. Wave server TCP/IP port information

Port	Direction	Communication description
20	Inbound and outbound	FTP (Active) port for populating the Wave service machines (WAVEWRKS, WAVEWRKL, WAVEWRKC) during the **Auto-Detect** process, and when you apply a fix pack (by using the doUpdate process).
21	Outbound	FTP port for populating the Wave service machines (WAVEWRKS, WAVEWRKL, WAVEWRKC) during the **Auto-Detect** process, and when you apply a fix pack (by using the doUpdate process).
22	Inbound (local only) and outbound	SSH for the following communication types: • Managed guests to get Linux, performance information, and function such as manage storage. • Communication back to itself to verify Wave server status.
111	Inbound	Remote Procedure Call (RPC) used by Network File System (NFS).
389	Inbound and outbound	389 is the default when you check **Enable user authentication through LDAP**, but you do *not* check **Use TLS from port 636**. To change the default, click **Administrative** > **Manage Parameters** > **Login** and change the port number.
443	Inbound	HTTPS. For the IBM Wave web launch page, GUI, and REST APIs.
514	Outbound	SYSLOG server port for audit logging.

Table 9. Wave server TCP/IP port information (continued)

Port	Direction	Communication description
636	Inbound and outbound	636 is the default when you check **Use TLS when connecting to LDAP (LDAPS://)** *and* **Enable user authentication through LDAP**. To change the default, click **Administrative** > **Manage Parameters** > **Login** and change the port number.
1952, 1953, 1954	Outbound	Communication to IBM service machines (WAVEWRS, WAVEWRKL, and WAVEWRKC) 1952 - 1954 are the defaults. To change the defaults, from the **Hardware viewer** right-click on the **z/VM System** > **Update Details**, and then change the **Service machine port** field.
2049	Inbound	Network File System (NFS).
3300	Inbound	Communication between the Wave GUI and the BTS. 3300 is the default. Any Wave application administrator can view the port, as shown in “BTS parameters” on page 117. Only a Wave server Linux administrator can change the port number. To change the port number *during installation*, use the procedure in step “8” on page 64. To change the port number *when updating Wave* (to a new fix pack, for example), use the procedure in “Upgrading IBM Wave for z/VM” on page 68.
3306	Inbound (local only)	Communication between the BTS and the database management system hosting IBM Wave's main database (see “The Wave Linux server (WAVESRV)” on page 8)
44444	Outbound	Communication to z/VM system by using SMAPI
55555	Outbound	Communication to z/VM system by using SMAPI

z/VM system ports

Table 10. z/VM system port information

Port Number	Direction	Description
20	Inbound and outbound	FTP (Active) port for populating the Wave service machines (WAVEWRKS, WAVEWRKL, WAVEWRKC) during the **Auto-Detect** process, and when you apply a fix pack (by using the doUpdate process).
21	Inbound	FTP port for populating the Wave service machines (WAVEWRKS, WAVEWRKL, WAVEWRKC) during the **Auto-Detect** process, and when you apply a fix pack (by using the doUpdate process).
23	Inbound	Telnet/TN3270 for 3270, or a CLC session. Port 23 is the default. To change the default port, in the **Hardware viewer** right-click on the **z/VM System** > **Update Details**, and then change the **3270 Connection Port** field.
1952, 1953, and 1954	Inbound	Communication to the IBM Wave service machines (WAVEWRS, WAVEWRKL, and WAVEWRKC). 1952 - 1954 are the defaults. To change the default port number, in the **Hardware viewer** right-click on the **z/VM System** > **Update Details**, and then change the **Service machine port** field.
44444, 55555	Inbound	Communication to the z/VM system by using SMAPI.
9999	Inbound and outbound	Communications for the Cross System Clone (CSC) service machines (WAVEWRKC).

z/VM Guest ports

Table 11. Managed guest port information

Port Number	Direction	Description
22	Inbound	SSH communication from the Wave Server to retrieve Linux and performance information, and functional such as manage storage. SSH communications from Windows for an SSH session (such as PuTTY).

Windows port information

Table 12. Windows port information

Port Number	Direction	Description
22	Outbound	SSH communication with the managed guests for an SSH session (PuTTY) to the guest.
23	Outbound	Telnet/TN3270 for 3270 or a CLC session
443	Outbound	HTTPS. For the IBM Wave web launch page, GUI, and REST APIs.
3300	Outbound	Communication between the Wave GUI and the BTS. 3300 is the default. Any Wave application administrator can view the port, as shown in “BTS parameters” on page 117. Only a Wave server Linux administrator can change the port number. To change the port number *during installation*, use the procedure in step “8” on page 64. To change the port number *when updating Wave* (to a new fix pack, for example), use the procedure in “Upgrading IBM Wave for z/VM” on page 68.

Firewall information

To ensure the end-to-end security of the IBM Wave solution, review “Port reference information” on page 77 with your network security administrator. Depending on your network topology, using a "white list" approach to firewall rules could also help to mitigate any possible risks. Guidance for setting up firewall rules might include the following:

- Open necessary ports only.

- Only allow client connections from intended users of the Wave GUI and the Wave APIs to the Wave server.
- Only allow client connections from z/VM guests managed by IBM Wave to the NFS server or servers configured in "NFS server usage" on page 45.
- Active mode FTP requires that the firewall allow inbound connections to the Wave server to unprivileged ports; only allow those connections from IP addresses associated with z/VM systems managed by Wave.

Chapter 3. Wave APIs and WebSphere Liberty

Wave provides a RESTful API via an embedded WebSphere Application Server Liberty. Using this server means that any HTTP client can interact with that server and issue API calls.

Wave requires that all API requests use a secure transport (HTTPS), include certain HTTP headers, and include IBM Wave user credentials. Wave can use existing IBM Wave user credentials to access the API, so you do not need to create specific users for this purpose.

API requests perform specific IBM Wave actions based on the associated Wave user's scope and permissions for the object that API request references. Wave users do not require specific scope or permissions for calling the API.

Wave embedded WebSphere Liberty

The Wave API is hosted on an embedded WebSphere Liberty application server. The WebSphere Liberty server resides in `/usr/wave/websphere`.

Note: You should avoid manually modifying the `server.xml` file; upgrades commonly replace its contents. If you find that you do need to manually update the `server.xml` file, you should make IBM aware of your requirement. When the upgrade process detects manual changes, it saves a copy of the pre-upgrade file in the `/usr/wave/client-cust` directory tree, but it does not try to replicate your manual modifications in the upgraded file.

Server configuration of the WebSphere Liberty server is done through the `server.xml` file. This file is located in the `/usr/wave/websphere/wlp/usr/servers/defaultServer/` directory. The `server.xml` file contains the following configuration settings:

1. Features that are applied in WebSphere Liberty.
2. The default user name used for displaying the documentation.
3. The HTTPS port number.
4. The logging level for the `console.log` file.
5. The SSL configuration.

For more information about authenticating IBM Wave API calls, see “Functionality parameters” on page 119.

The API calls are issued by using HTTPS on the port specified in the `server.xml` file. This port defaults to 443. For more information, see Table 9 on page 77.

The IBM Wave API server produces logs in various locations.

`/var/log/WAVE`
Among other logging information, this folder contains files with the prefix `APILog`. Those files contain all the logging information for IBM Wave API calls. The amount of information can be changed by setting properties in `/usr/wave/API/user_config.properties`.

`/usr/wave/websphere/wlp/usr/servers/defaultServer/logs`
This folder contains `console.log` and `messages.log`, which are log files that are generated by the WebSphere Liberty server. The amount of logged information can be changed by setting properties in `/usr/wave/API/user_config.properties` file.

For more information, see “Wave server log options” on page 134.

Starting and stopping the WebSphere Liberty server

The Wave RESTful API is hosted on an embedded WebSphere Application Server Liberty installation. Wave's installation and upgrade process starts this server automatically as a Linux service, and configures

Linux to start it automatically after reboot. In general, you should not have to take any manual actions for this service.

Stopping, starting, and querying the server status is done by running the following commands on the WAVESRV command line:

- To query the status of the server, enter:

```
systemctl status WAVEWebServer
```

- To start the server, enter:

```
systemctl start WAVEWebServer
```

- To stop the server, enter:

```
systemctl stop WAVEWebServer
```

The service is enabled and started automatically by `systemctl` on boot.

Chapter 4. Administrative actions

The administrative information covers all of the **Administrative** actions that are available from the **IBM Wave Main Menu** > **Administrative**.

Site Management

The **Administrative** > **Site Management** menu contains the options to work with the management aspects of the IBM Wave for z/VM.

External Entities Manager

Before IBM Wave can manage a z/VM system or other object, the entity must be defined as an external entity.

To open the **IBM Wave External Entities Manager** window, from the IBM Wave main menu, select **Administrative** > **Site Management** > **External Entities Manager**.

Using the **External Entities Manager**, you can add, update, and remove such IBM Wave managed entities as a z/VM system, a router, or storage controller. Each managed entity is marked with an icon that illustrates its type.

Right-click on a managed entity in the table, as shown in Figure 30 on page 85, to **Display Details**, **Update Details**, or **Remove**.

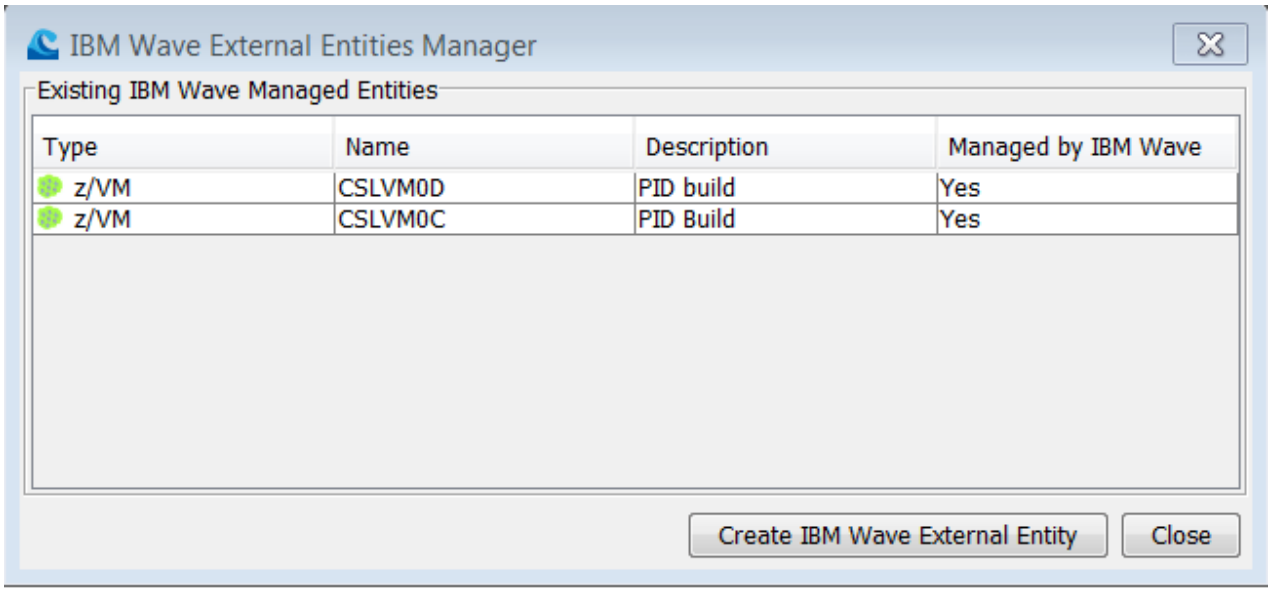


Figure 30. IBM Wave External Entities Manager window

Creating a new external entity

To add a new z/VM or z/OS system, router, or controller, from the **IBM Wave External Entities Manager**, click **Create IBM Wave External Entity**. Select the type of managed entity from the menu, and then fill in the name of the entity (such as the z/VM system name as shown in Figure 31 on page 86). You can optionally add the description of the managed entity. Click **Create** to add the new entity to Wave.

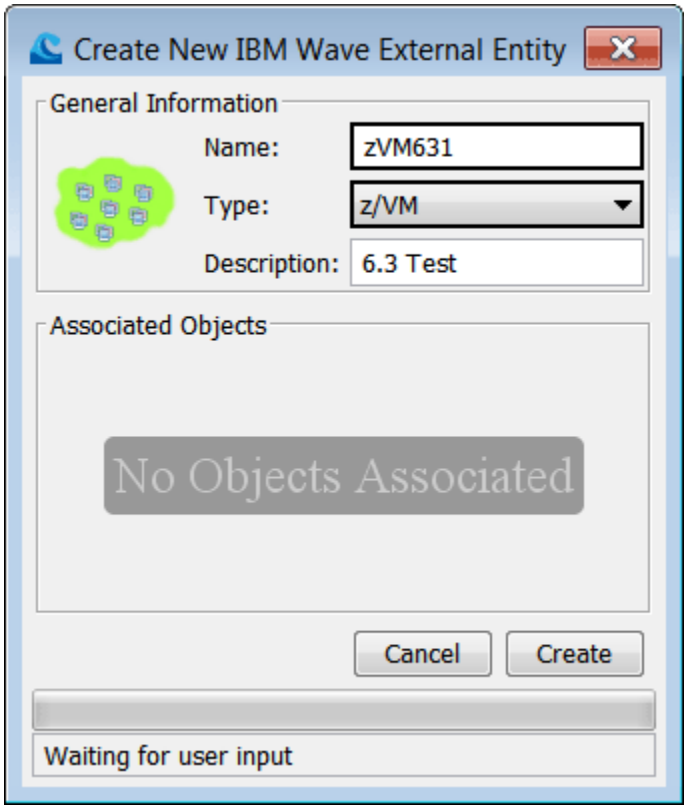


Figure 31. Create New IBM Wave External Entity

For more information, see Create a new external entity.

Manage Device Pools

Use the **Manage Device Pools** option to add, update, and remove device pools from the IBM Wave database.

To open the **Device Pool Manager**, from the IBM Wave main menu, select **Administrative** > **Site Management** > **Manage Device Pools**. Through the **Device Pool Manager**, you can add, update, and remove device pools from the IBM Wave database. Right-click on an entry in the table to complete the following tasks

- Display details
- Update details
- Update an IAN
- Remove the device pool.

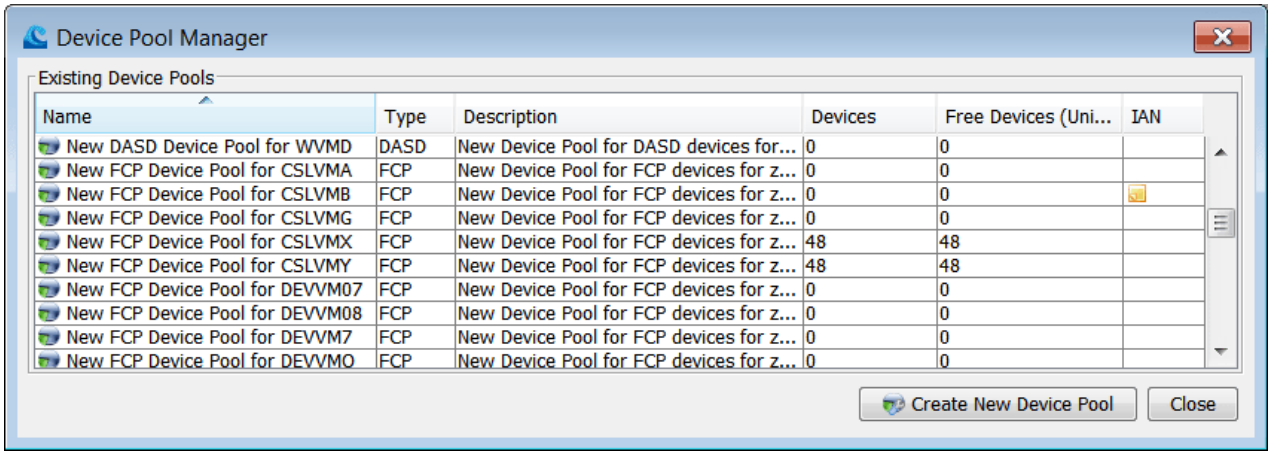


Figure 32. Device Pool Manager

For more information about device pools, see Device pool functions.

Manage Virtual Network Segments

Use the **Virtual Network Segment Manager** to add, update, and remove Virtual Network Segments from the IBM Wave database.

To open the **Virtual Network Segment Manager**, from the IBM Wave main menu, select **Administration** > **Site Manager** > **Manage IBM Wave Virtual Network Segment**.

Right-click on a table entry to get the menu with the display, update, and remove actions. For more information about actions and tasks for virtual network segments, see Virtual network segment functions.

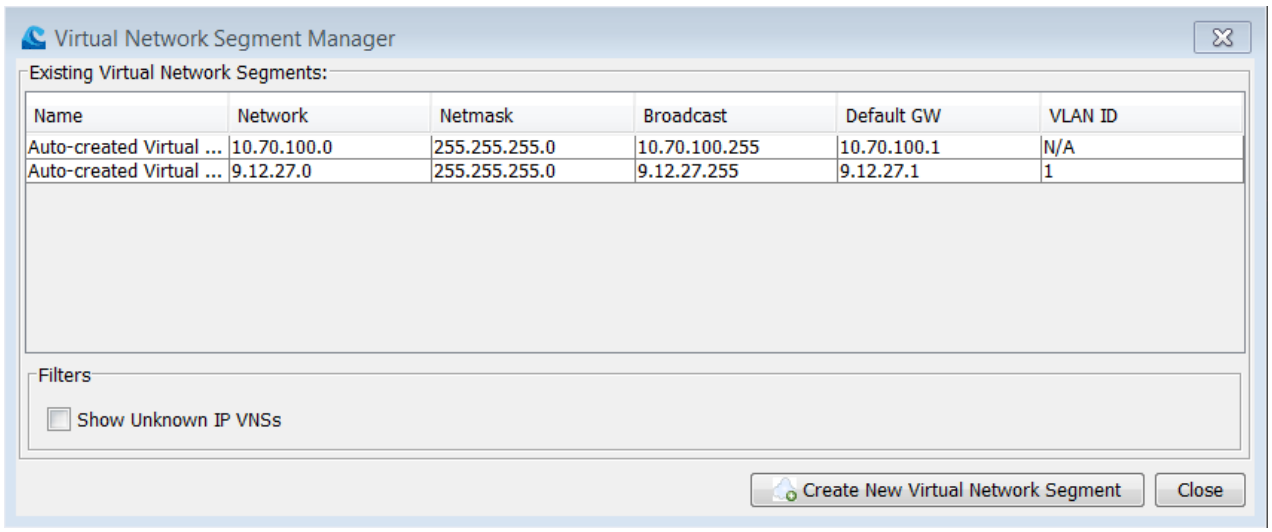


Figure 33. Virtual Network Segment Manager

Custom Attribute Manager

Use the **Custom Attribute Manager** to define your own custom attributes.

To open the **Custom Attribute Manager**, from the IBM Wave main menu click **Administrative** > **Site Management** > **Custom Attribute Manager**.

Using the **Custom Attribute Manager**, you can add, edit, and remove custom attributes and the respective possible and default values. The **Existing Attributes** pane contains a list of all the defined custom attributes. When a custom attribute is selected, the attribute properties are displayed in the Attribute Details pane and the Attribute Values pane.

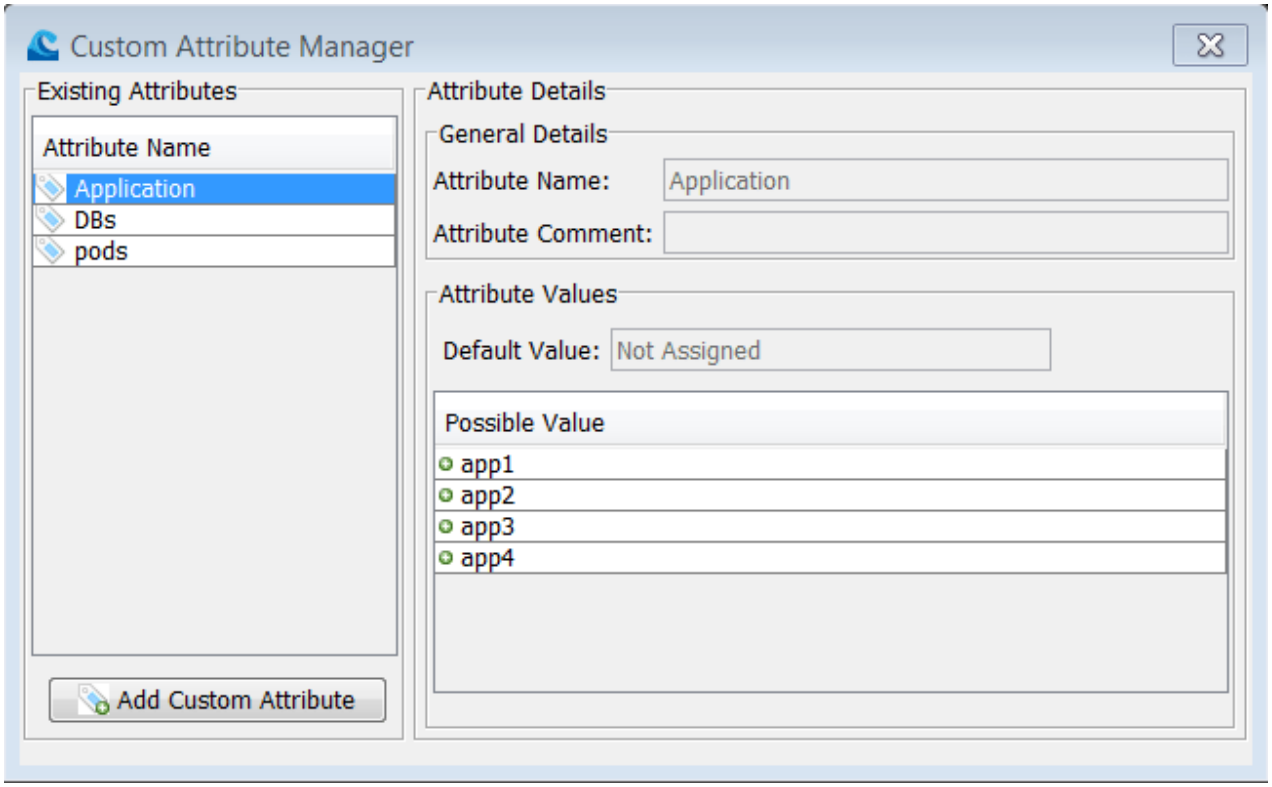


Figure 34. Custom Attribute Manager

For more information and instructions about assigning and working with custom attributes, see "Custom attributes" on page 17.

z/VM Directory Manager

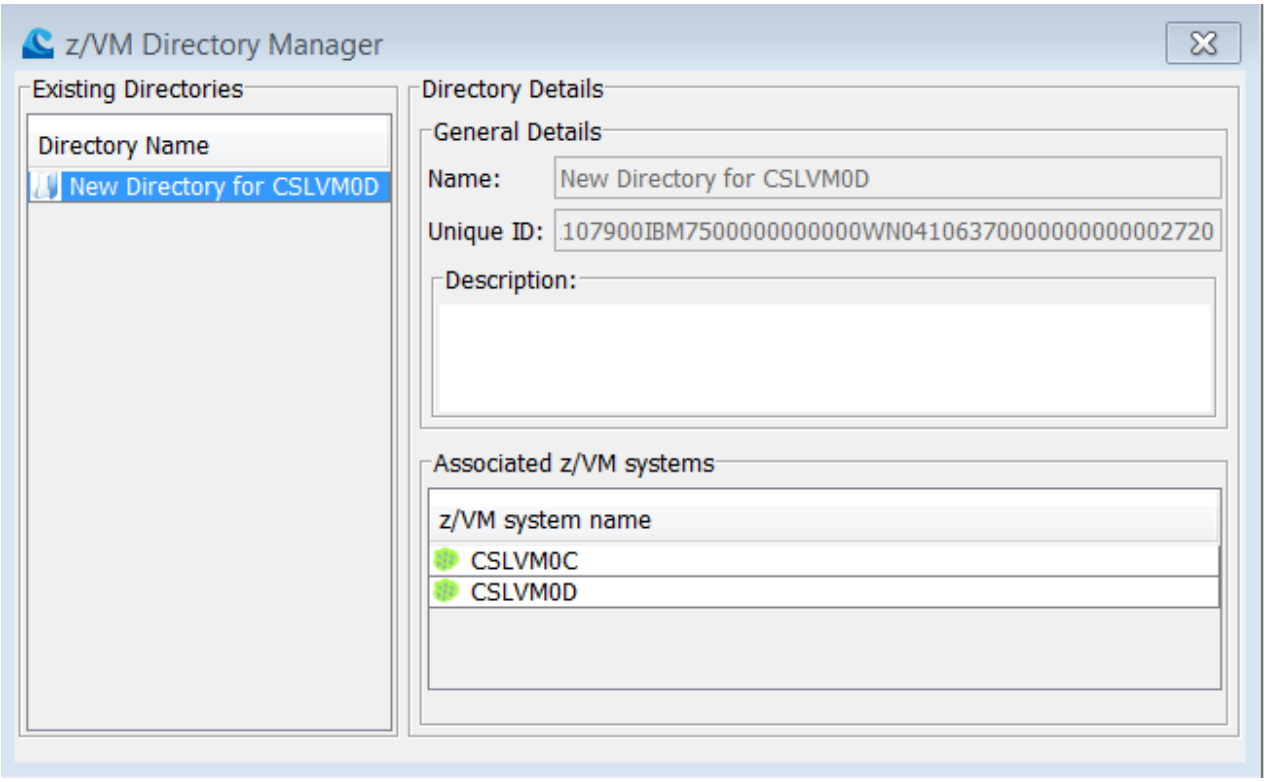
Use the **z/VM Directory Manager** to view and interact with z/VM directories that are discovered by IBM Wave.

To open the **z/VM Directory Manager**, from the IBM Wave main menu, click **Administrative** > **Site Management** > **z/VM Directory Manager**.

When you open the **z/VM Directory Manager**, you can view and interact with the z/VM directories that are discovered by IBM Wave. The left pane displays a list of the names of all the discovered directories. When you select a directory from the left pane, the right pane displays information about the directory such as all of the z/VM Systems that are managed by IBM Wave and the:

- Name
- Unique ID
- Description
- Associated z/VM Systems.

To update the information for a directory, right-click on the directory in the left pane and select "Update". You can update the directory's name and description.



For information about how to update the "Unique ID" for the z/VM system directory, see “z/VM directory unique ID changes” on page 89.

z/VM directory unique ID changes

If you move your source directory disk from one direct access storage device (DASD) volume to another, the unique identifiers (unique ID) for the z/VM system directory changes. You must use the auto-detect process to update IBM Wave to handle the directory changes.

The reasons for changing the DASD volume for your source directory can vary. For example, when you migrate a storage controller it changes the physical DASD on which the directory is stored, and affects the unique ID. If you do not update IBM Wave to reflect the directory changes with the new unique ID, it can block new systems from being added and cause systems to be associated with the wrong directory.

To migrate your source directory disk from one DASD volume to another, see “Changing the source directory” on page 89.

Changing the source directory

Use the following procedure when you must migrate your source directory disk from one DASD volume to another.

Procedure

1. Suspend all the z/VM systems that are associated with the original source directory.
2. Select one z/VM system on which to run the auto-detect process. From the IBM Wave main menu, select **Auto-Detect**.

   The new directory is automatically created and the unique ID (metadata) is copied into the new directory.
3. After the auto-detect process completes, you must repeat the process for each system that is associated with the original directory.

   Each system is automatically moved to the new directory with the metadata intact.
4. After all of the systems are successfully migrated, open the directory manager and delete the original directory.

   There must no longer be any z/VM systems associated with the original directory.

Results
All of the z/VM systems now appear in the new directory with the appropriate metadata.

z/VM Account Manager

Use the **z/VM Account Manager** to manage known z/VM Accounts in all managed z/VM Systems.

All of the existing z/VM Accounts are listed in the **Existing z/VM Accounts** table. Select an account to display its details in the **Account Details** pane.

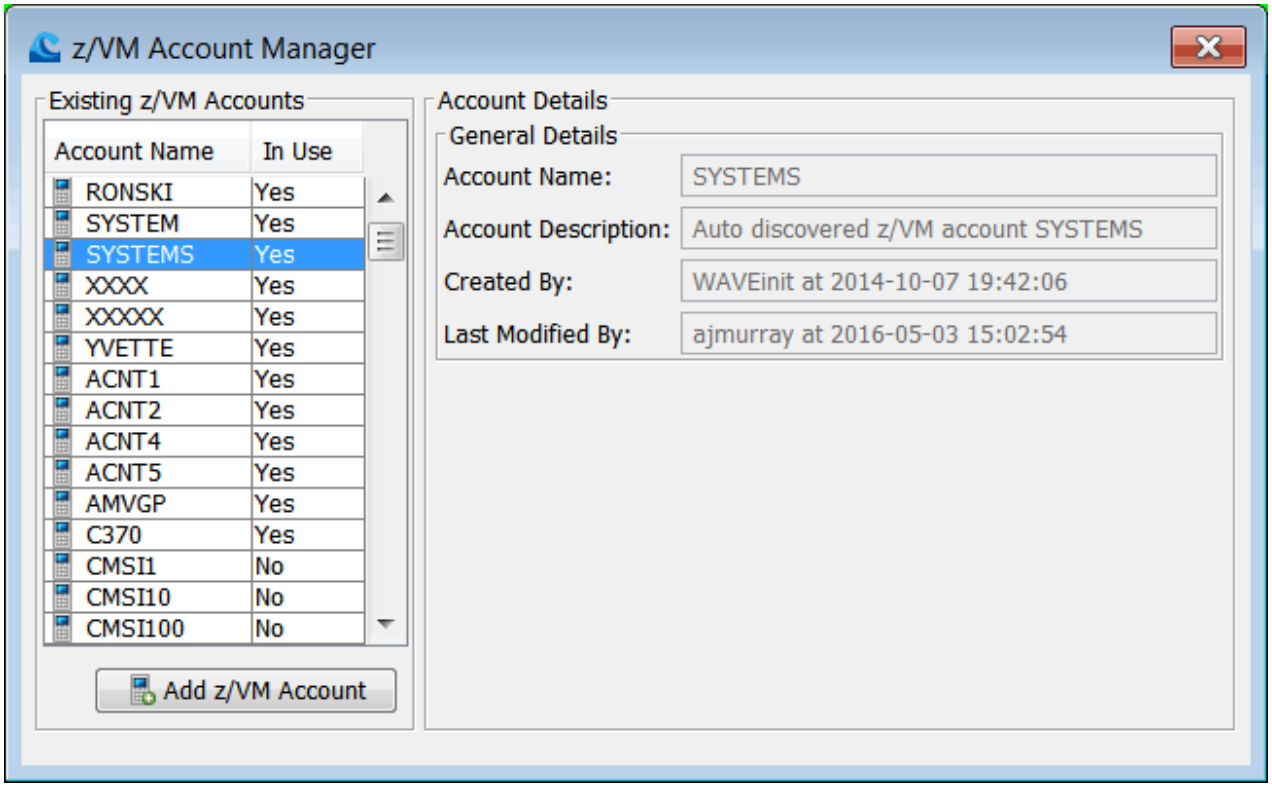


Figure 35. z/VM Account Manager

Right-click on an account to complete the following tasks:

- **Update z/VM Account** - You can update the account description for any account.
- **Delete z/VM Account** - You can delete an account that is not in use.

For more information about adding, deleting or updating an account, see "Add, Update, or Delete z/VM accounts" on page 90.

Add, Update, or Delete z/VM accounts
Use the **z/VM Account Manager** add, update, or delete z/VM accounts.

To create a new z/VM account, click **Add z/VM Account**, and then enter the account name and description.

To update the descriptive fields for a z/VM Account, right-click the account in the **Existing z/VM Accounts** pane, and then click **Update z/VM Account**.

To delete an account, right-click the account in the **Existing z/VM Accounts** pane, and then click **Delete z/VM Account**.

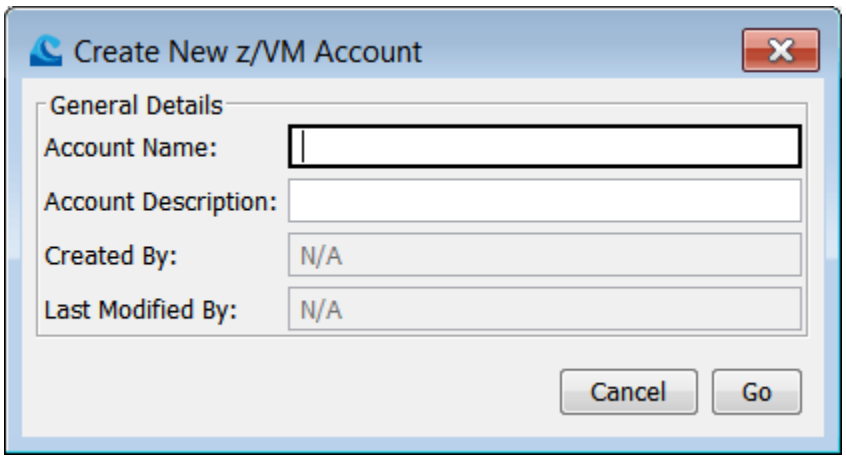


Figure 36. Add z/VM Account

The **Create New z/VM Account** menu contain the following "General Details":

- **Account Name** - The required name of the z/VM Account.
- **Account Description** - The optional description for the z/VM Account.
- **Created By** - The user or process that created, modified, or automatically detected of the z/VM Account and the time stamp.
- **Last Modified By** - The user or process to last update the z/VM Account and the time stamp for that update.

Notes:

1. When you update or create a z/VM Account, it has no effect on any of the managed z/VM Systems, but other actions can impact the z/VM System. Using the new or updated z/VM Account from the **Assign z/VM Account** multiple task action for one or more guests, changes the ACCOUNT directory statement for the selected guests.
2. You cannot delete a z/VM Account if it is assigned to a z/VM Guest. If the guests to which the z/VM account is assigned to are outside of the user's scope, IBM Wave issues a message.

AGC Manager

Use the Automatic Guest Classification (AGC) Manager to create, update, and delete AGC Entries.

To access the **AGC Manager**, click **Administrative** > **Site Management** > **AGC Manager**.

Important: After AGC is enabled in the **IBM Wave Parameters**, the AGC rules are strictly enforced. For complete information, see all of the AGC information in “Automatic Guest Classification” on page 37.

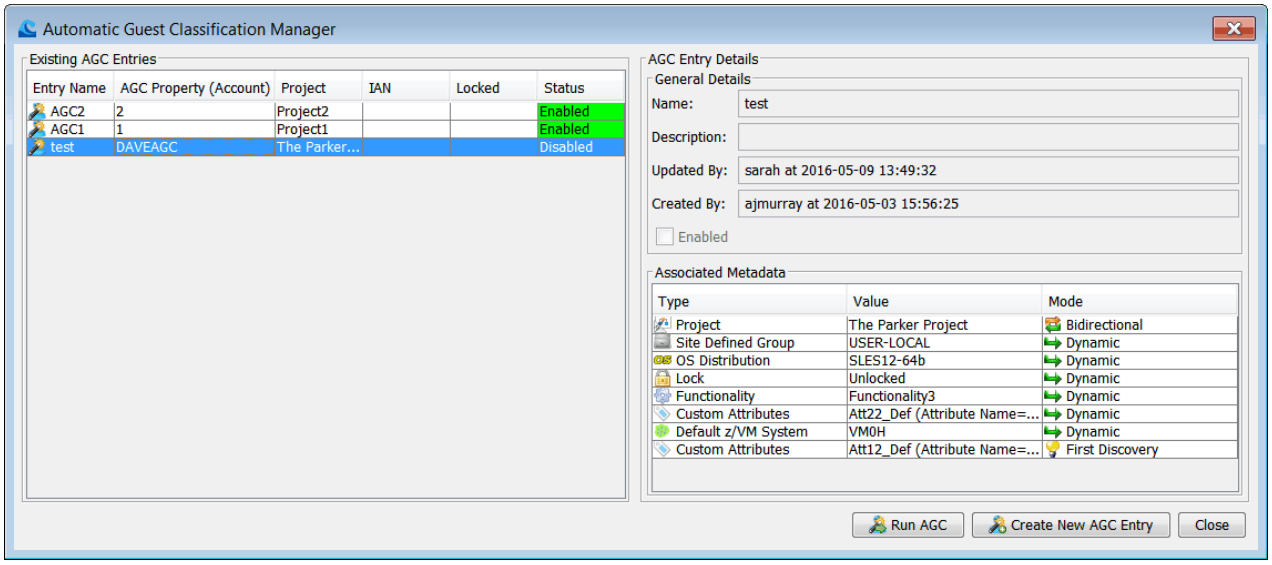


Figure 37. Automatic Guest Classification Manager

Note: A warning message at the top of the window indicates when AGC is not in use.

Related tasks

"Running Automatic Guest Classification (AGC)" on page 43
Running Automatic Guest Classification (AGC) associates IBM Wave metadata elements with the z/VM Guest's directory entry.

Related information

"Automatic Guest Classification" on page 37
"AGC Manager" on page 37
"Defining AGC entries" on page 41
"Resolving AGC conflicts and inconsistencies" on page 44

Update Authorized TVP-API Credentials

Use the **Update Authorized TVP-API Credentials** to update the target virtualization platform (TVP) API user ("Authorized API User").

To open the **Update Authorized TVP-API Credentials** window, from the IBM Wave main menu, select **Administrative** > **Site Management** > **Update Authorized TVP-API Credentials**.

Use the **Update Authorized TVP-API Credentials** menu to change the authorized TVP-API user name and password for one or more target virtualization platforms (TVP) in the database. The top pane contains a field for the new user name and a field for the password value to set for the TVP-API credentials. The bottom pane contains a list of the systems that are defined in IBM Wave. The following options are available:

- **Update** - When you click **Update**, the "Username" and "Password" value is applied for any TVP that is checked in the **Update Authorized TVP-API Credentials for z/VM Systems** pane.
- **Parallel** - When you click **Parallel**, the operation can run on multiple TVPs in parallel.
- **Go** - When you click **Go**, a BTS request is generated to update the authorized TVP-API credentials.

When you update the authorized TVP-API credentials, IBM Wave ensures that a user with the specified user name exists on the TVP and that the password matches the specified password. If both conditions are not met, the individual BTS request that is created for the operation terminates with no change to the affected TVP and the status of the BTS work unit is `Error`.

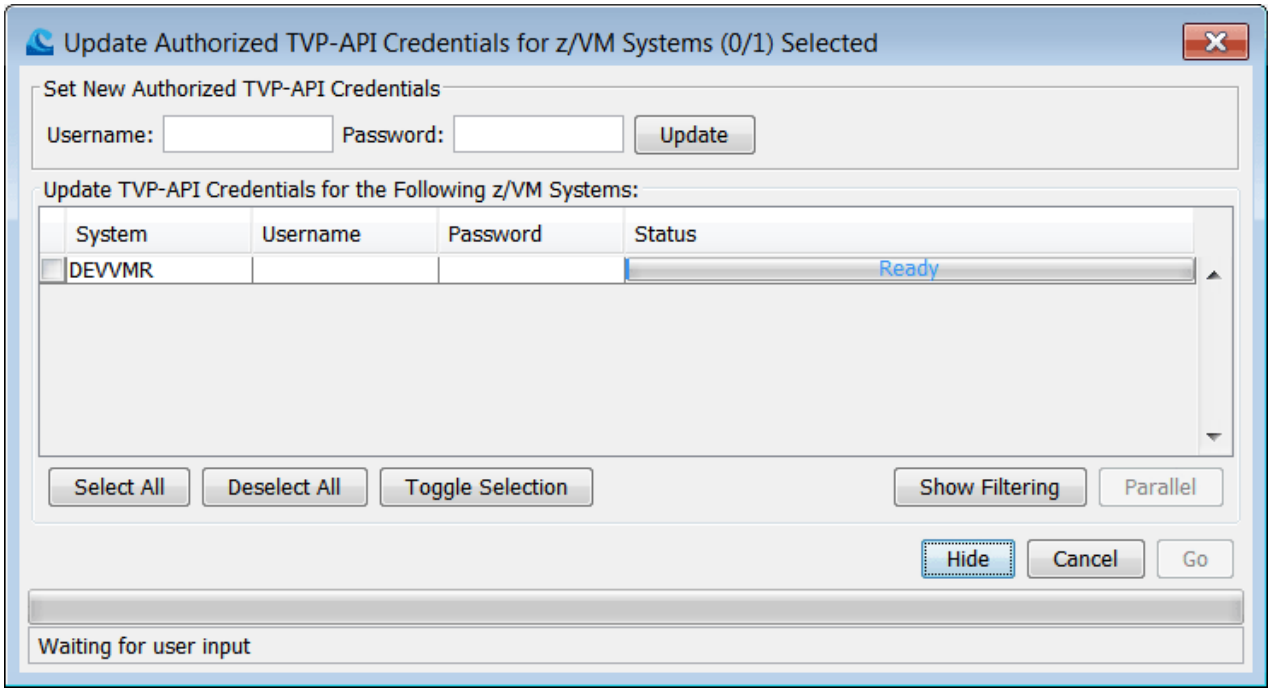


Figure 38. Update Authorized TVP-API Credentials

Fields in this pane:

- **Username** - The name of the authorized TVP-API user.
- **Password** - The security password for the user name. Mouse (or hover) over this field in the GUI to see information about password length and valid characters.

Precautions for changing the Authorized TVP-API password on the z/VM TVP

The system administrator can change the password of an authorized Target Virtualization Platform (TVP) API user on z/VM. It is important to take precautionary steps to maintain the integrity of TVP API credentials, which are stored in the IBM Wave database. Before IBM Wave permits an administrator to change the password for the TVP API user, the managed z/VM system must be suspended in IBM Wave.

1. Before changing the password, suspend the z/VM system and all outstanding tasks on the BTS tab.
2. As a precautionary measure, before changing the password, the administrator must ensure that all outstanding scheduled tasks on the BTS are permitted to complete or are terminated. For more details about viewing scheduled BTS tasks through the BTS Manager, see the topic about the ""BTS Manager" on page 106".
3. If the z/VM system is a member of an single system image (SSI) cluster with a shared directory, or a shared RACF database, or both, take the same action for every z/VM system in the cluster.
4. Change the password.
5. After the password is changed for the guest on z/VM, update the TVP API credentials to reflect the new password. If the z/VM system is a member of an SSI cluster with a shared directory, a shared RACF database, or both, repeat this action for every z/VM system in the cluster.

Resume the z/VM system(s) in IBM Wave. For information about how to suspend and resume a z/VM TVP in IBM Wave, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_update_details1.html.

Update Minidisk Passwords

Use the **Update Minidisk Passwords** action to change the service machine passwords in the IBM Wave database to match the passwords in use on z/VM.

To open the **Update Minidisk Passwords** window, from the **IBM Wave Main Menu**, click **Administrative** > **Site Management** > **Update Minidisk Passwords**.

Note: The passwords must match the passwords in use on z/VM.

To change the passwords, enter the new password in the **Read**, **Write**, and **Multi** fields. Click **Update** to change the passwords for one or more systems listed in the table. To make the password changes in the IBM Wave database, click **Go**.

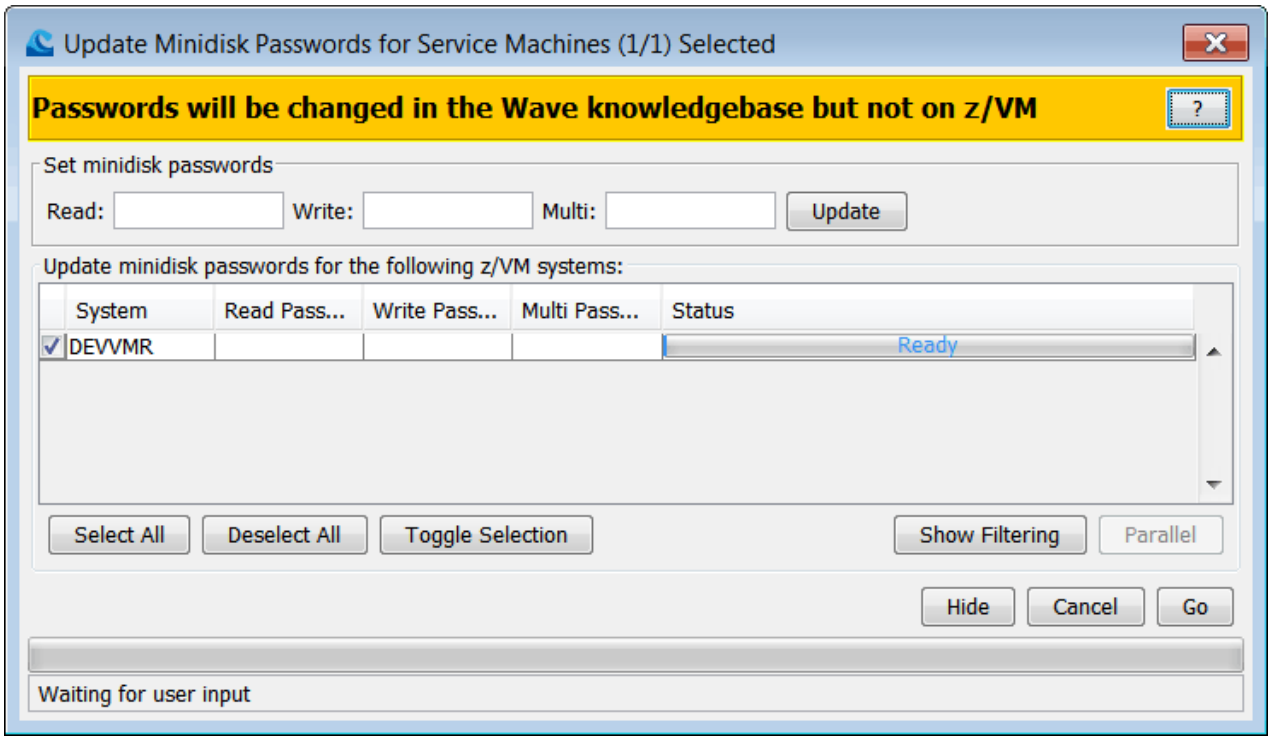


Figure 39. Update Minidisk Passwords

The following fields are in the **Set Minidisk Passwords** pane:

Read
The **Read** password for the minidisk in the IBM Wave database.

Write
The **Write** password for the minidisk in the IBM Wave database.

Multi
The **Multi** password for the minidisk in the IBM Wave database.

Manage FCP Information

Use the **Manage FCP Information** option to associate WWPNs with storage controllers.

To manage FCP information, from the main menu, click **Administrative** > **Site Management** > **Manage FCP Information**. Using the **FCP Manager**, target WWPNs that are detected by IBM Wave can be associated with Storage Controllers. To change the Storage Controller associated with a specific target WWPN, right-click the WWPN entry and select "Assign To Storage Controller".

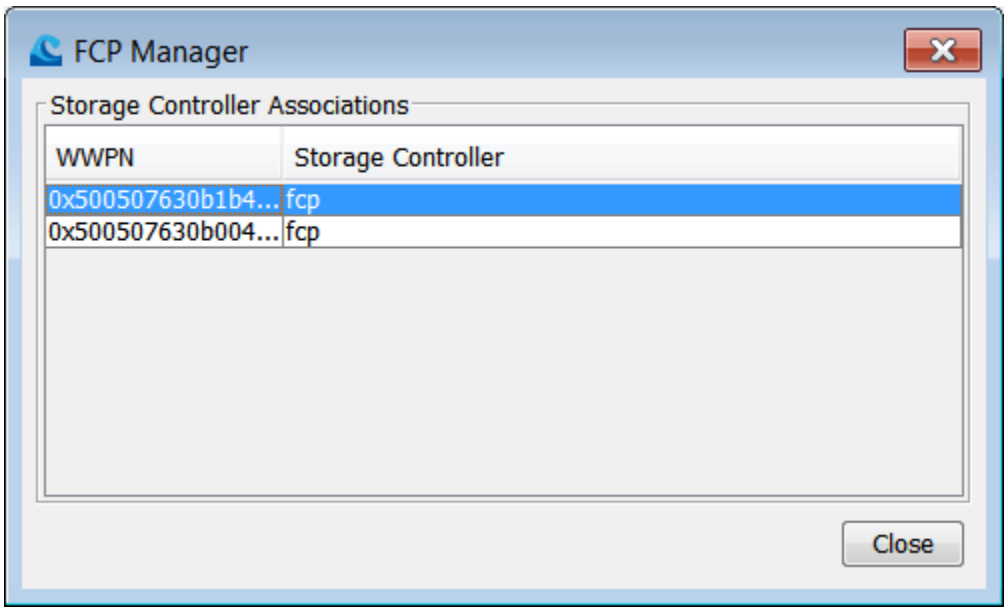


Figure 40. FCP Manager

Note: Assigning Target WWPNs to Storage Controllers is not mandatory, and can be used for documentation purposes. When using the **Storage Chart** feature, the diagram displays the storage controllers based on the associations.

IBM Wave Linux Repository Manager

Use the **IBM Wave Linux Repository Manager** to add Linux media to the IBM Wave Database, or work with and view the status of existing Linux media.

Using the **IBM Wave Linux Repository Manager**, you can install new versions of Linux media and view the existing Linux repositories. From the **IBM Wave main menu**, select **Administrative** > **Site Management** > **IBM Wave Linux Repository Manager**.

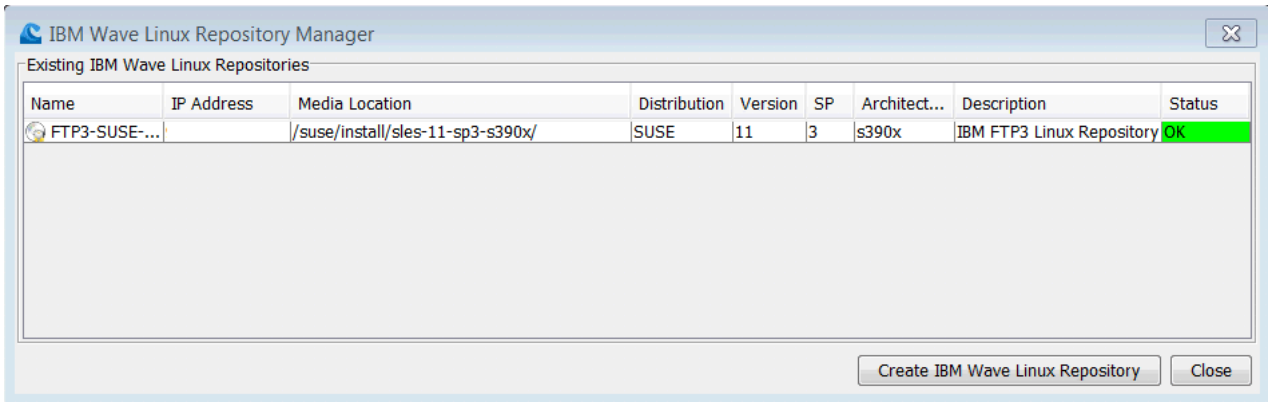


Figure 41. IBM Wave Linux Media Repository Manager

The **IBM Wave Linux Repository Manager** window displays all known Linux media repositories with the following information:

- **Name** - The name of the Linux Media Repository.
- **IP Address** - The IP address of the server that holds the repository.
- **Media Location** - The location on the server where the Linux media repository is installed. The location is typically 70 characters or less, but for Ubuntu, the media location cannot be longer than 58 characters. Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES) have rules about the parameter and configuration files. For more information about the parameter and configuration files, see the installation documentation for the Linux versions in your installation.

- **Distribution** - The Linux distribution for the repository (for example, SLES, RHEL, Ubuntu).
- **Version** - The version of the Linux installation material that is stored in the repository.
- **SP** - The Service Pack of the Linux installation media that are stored in the repository.
- **Architecture** - The architecture (such as s390, s390x, i386) of the Linux installation media that are stored in the repository.
- **Description** - The description that is given to the Linux Media Repository.
- **Status** - Indicates the status of the repository. Under certain circumstances, the IBM Wave database can become out of sync with the repository. A message that indicates the status of the repository.

For more information about Linux media repositories, see “IBM Wave Linux media repository” on page 46.

Adding, updating, and deleting Linux media repositories

To create a new Linux media repository, go to the **IBM Wave Linux Repository Manager**. From the IBM Wave main menu, select **Administrative > Site Management > IBM Wave Linux Repository Manager**. In the **IBM Wave Linux Repository Manager**, click **Create IBM Wave Linux Media Repository**.

Right-click on the table row to **Display or Update Details** for the selected Linux media. Click **Update Details** to update descriptive fields for the media, or click **Remove** to delete the Linux instance from the IBM Wave Linux Repository (the instance is also removed from the IBM Wave Database).

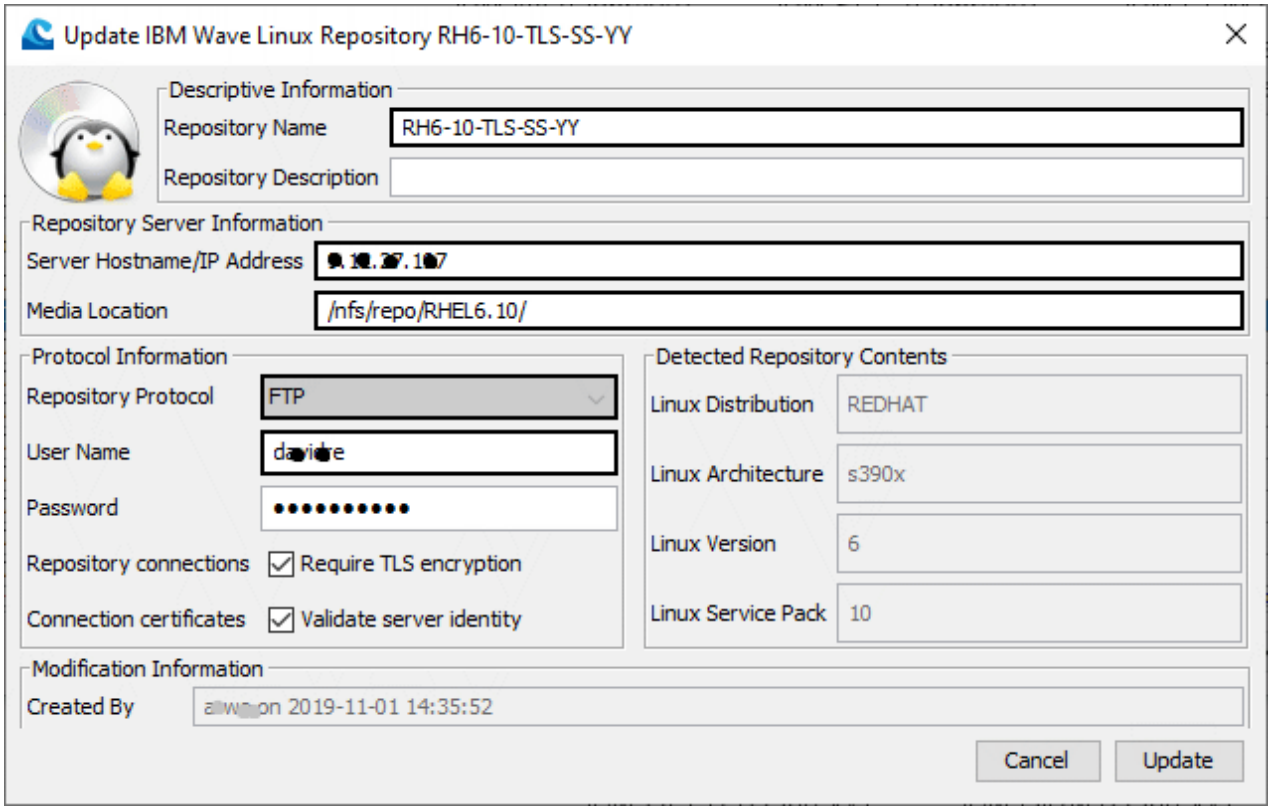


Figure 42. Add or update details for an IBM Wave Linux Repository

When you add or update IBM Wave Linux media repositories, a window appears with the following fields:

- **Repository Name** - The name of the repository. This can be any name you choose.
- **Repository Description** - An optional description of the repository.
- **Server IP Address** - The IP address of the server where the repository resides.
- **Media Location** - The location on the server where the repository resides.
- **Repository Protocol** - The communication protocol used to access the repository. Currently, only the FTP and HTTPS protocols are supported.

- **User Name** / **Password** - The credentials used to access the repository, using the repository protocol. The password is not required to create a repository.
- **Repository connections - Require TLS encryption** - This field governs connections from the Wave server (acting as a client) to the server hosting Linux installation files and (during installation) from the Linux installation program to the same server. (For more information, see Installing Linux with the BMI wizard.)

  When the box is checked (the more secure option), TLS is required to establish all connections. When you create a new repository, the default is to require a secure connection.

  When the box is not checked, which Wave only permits when FTP is selected as the protocol, Wave will attempt secure FTP connections, and will fall back to insecure FTP connections only if it can't establish a secure FTP connection.
- **Connection certificates - Validate server identity** - This check box controls whether or not the server's certificate must be valid in order to connect securely; it has no meaning when server connections are unencrypted.

  When the box is checked (the more secure option), the server's server certificate must be seen as valid by each client (Wave's server and the Linux installation program) in order to connect securely; if certificate validation fails, no connection is established. This might cause some operations to succeed, while others fail (if the server's certificate is signed by an internal enterprise certificate authority that Wave has been configured to trust to sign certificates, but the Linux installation program does not trust that CA, for example).

  When you create a new repository, the default is that the server's server certificate must be valid in order to establish an encrypted connection.

  When you select the HTTPS protocol, the URL's host name or IP address must match one of the SubjectAlternativeName values in the server's certificate; no such check occurs by Wave when you select FTP as the protocol.

  When the box is unchecked, validity checking of the server's server certificate is skipped.
- **Linux Distribution / Linux Architecture / Linux Version / Linux Service Pack** - These fields, which are detected automatically when the Linux media repository is added, indicate the various parameters of the Linux installation media located in the repository.
- **Created By** - Indicates the IBM Wave user, the date and time when the repository was created, and when it was last modified.

Linux media repository creation processing

When creating a new IBM Wave Linux media repository, a Background Task Scheduler (BTS) work unit is sent to the BTS. The work unit is comprised of several BTS requests that perform the following actions:

1. **Linux Media Repository Discovery** - IBM Wave attempts to access the repository using the selected protocol. After the repository is accessed, IBM Wave automatically detects the content of the repository and updates the distribution, version, architecture, and service pack descriptive fields in the IBM Wave database. A failure to discover the Linux media repository causes the process to fail.

   For example, when IBM Wave discovers the Red Hat repositories, it attempts to detect information by reading the `.treeinfo` file from the root directory of the Red Hat installation media. Because of the hidden nature of the `.treeinfo` file, the file can sometimes be missed. Ensure that the `.treeinfo` file gets copied over into the Linux directory.
2. **Add the Linux Media Repository to the IBM Wave Database** - After the discovery task is complete, IBM Wave adds the repository to the database. As part of the process, a default minidisk address is allocated for the repository. The minidisk address is used when creating the dedicated minidisk on the Short Service Machine during the **Launch Linux Installation** action.

For more information, see Installing Linux with the BMI Wizard.

Linux media repository update processing

Depending on which fields are updated, the following actions occur:

- If any of the following fields are updated, a discovery task is submitted to the Background Task Scheduler (BTS):
  - **Server IP Address**
  - **Media Location**
  - **Repository Protocol**
  - **User Name** / **Password**
  - **Repository connections**
  - **Connection certificates**
- Regardless, a task is submitted to update the IBM Wave database with the new information.

  Note: When the discovery task fails, the task is skipped.

Linux media repository delete processing

When you delete a Linux media repository, special processing occurs if the Linux media repository was used in one or more managed z/VM systems as part of the **Launch Linux Installation** action.

If the repository was used, IBM Wave attempts to remove the dedicated minidisk for the short service machine of all the z/VM systems in which the Linux media was used. If IBM Wave is unable to remove the dedicated minidisk from one or more of the short service machines, it is indicated in the COR output of the request and the status of the repository. A rediscovery of the repository is necessary to resolve the issue.

If the process completes successfully, the Linux media repository is removed from the IBM Wave database.

If the Linux media repository was not used, it is removed from the IBM Wave database.

Add New CPC

Use **Add New CPC** to add a new central processor complex (CPC) to IBM Wave.

To add an IBM Z® or IBM LinuxONE CPC to IBM Wave, you can choose from the following options:

- From the main menu click **Administrative** > **Site Management** > **Add New CPC**.
- In the **Hardware Viewer**, right-click in the white space, and then click **Add New CPC**.

After you install IBM Wave, you can define a CPC by providing a following attributes:

- **CPC name** - A name that uniquely identifies the CPC.
- **CPC model** - For example, 3096.
- **CPU ID** - The ID that validates that IBM Wave is licensed to run on the system.

To obtain the information, run the CP command **`QUERY CPUID`**. The command returns the following response:

```
CPUID = aasssssscccddddd
```

The ssssss string (hexadecimal digits three through eight) is the value for the CPC that you must enter in IBM Wave.

For more information about all the CPC actions, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_hardware_functions1.html.

Remove CPC

Use **Remove CPC** option to remove a central processor complex (CPC) from IBM Wave.

To remove a CPC from IBM Wave, from the main menu click **Administrative** > **Site Management** > **Remove CPC**. From the menu, select the CPC that you want to remove.

After the **Remove CPC** window loads (as shown in Figure 43 on page 99), click **Remove** to remove the system from IBM Wave management.

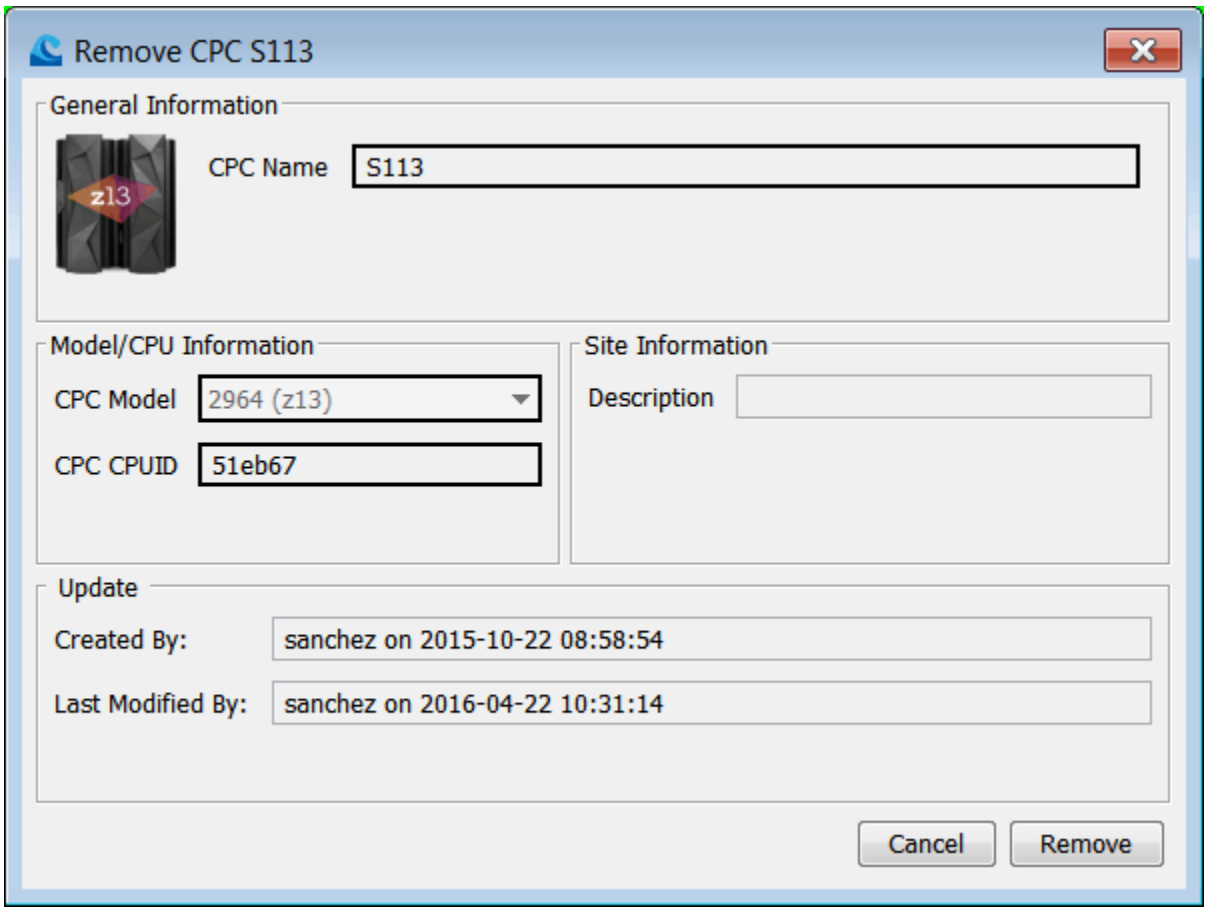


Figure 43. Remove a CPC

Tools

Use the **Tools** > **Import Guest Information** option to import guest metadata from a .csv file.

The **Administrative** > **Site Management** > **Tools** option contains the **Import Guest Information** option. For more information, see "Import Guest Information" on page 99.

Import Guest Information
Use the **Import Guest Information** action to import guest attributes that you want to apply in IBM Wave for z/VM.

To open the **Import Guest Information** window, from the **IBM Wave Main Menu**, click **Administrative** > **Site Management** > **Tools** > **Import Guest Information**. You are prompted for the location of a comma-separated value (.csv) file that contains a list of guest attributes to import.

For guidance, details, and an example, see "Import guest metadata" on page 18.

Toggle Single User Mode

When applying maintenance to IBM Wave, consider using the single user mode option.

To turn the **Toggle Single User Mode** on or off, select the **Administrative** > **Toggle Single User Mode**. **Toggle Single User Mode** is required for certain IBM Wave maintenance functions. When other users are logged on, IBM Wave provides an option to send a message to the active users to ask them to log off. There's also an option to force users to log off when necessary.

IBM Wave Database Actions

To use the database actions, from the IBM Wave main menu, select **Administrative** > **IBM Wave Database Actions**. You can select the following tasks:

This section also explains "Restoring the IBM Wave database" on page 100.

Set IBM Wave Database Backup Password

IBM Wave requires that you protect your database backups with a password. This same password will protect all of its database backups, whether scheduled or initiated manually, and must be supplied to the Wave server's Linux administrator if you ever need to restore the database from a backup. If IBM Wave Support ever requests a database dump as part of diagnostic data collection, the dump is protected by the same password, and would need to be supplied to Wave Support.

To set this password, from the IBM Wave main menu, select: **Administrative** > **IBM Wave Database Actions** > **Set IBM Wave Database Backup Password**.

Important: You should save the password value in a secure place outside of IBM Wave. There is no way to display the current password value, and no way to recover it. If you forget the password, backups secured by it will be permanently inaccessible; in this case, you should set a new password as described above and create a new backup as described in "Backup IBM Wave Database" on page 100. If you enabled scheduled backups in the past, those backup tasks will fail until you set a password for them.

Backup IBM Wave Database

To create a backup copy of the IBM Wave database, from the IBM Wave main menu, select **Administrative** > **IBM Wave Database Actions** > **Backup IBM Wave Database**.

The backup file for the IBM Wave database is immediately written to a file on the WAVESRV server that is encrypted and stored in the `/usr/wave/DBBackup` directory. The backup file name contains a unique date and time stamp, which means the file cannot be overwritten by a subsequent backup (as shown inFigure 44 on page 100):

```
backupSQLDump-<Day>-<month>-<Day of Month>-<Hour>-<minute><Second>-<Year>.wavedb
```

Figure 44. Backup file name format

Important: By default, the database backup task is inactive. To activate it, update the task by using the **Schedule Parameter** cell in the Background Task Scheduler (BTS). For information about updating and scheduling a periodic backup task, see the **BTS: Scheduling Tab** in "BTS Manager" on page 106.

To help control the amount of data that is stored on the server by the backup process, you can create a cron job. In the following example, the five most recent database backups are kept in the `/usr/wave/DBBackup` directory, and all others are removed:

```
ls -tr /usr/wave/DBBackup/backupSQLDump*.wavedb | head -n -5 | xargs --no-run-if-
empty rm
```

Important: Installing a new RPM clears `/usr/wave` and its subdirectories. When you plan to install a new build and restore from the backup version, ensure that you move the backup file out of the `/usr/wave` directory.

Restoring the IBM Wave database

When you restore the IBM Wave database from a backup copy, you are replacing the current IBM Wave configuration with the backup version.

Before you begin

Before you proceed, save a backup of the current database configuration. To create a backup of the IBM Wave database, see "Backup IBM Wave Database" on page 100.

About this task

You must have root credentials to use SSH to access the file system. To restore the IBM Wave database, you need a backup version (that you created by using the instructions in "Backup IBM Wave Database" on page 100).

Procedure

1. Exit the IBM Wave client.
2. Log in to the WAVESRV server by using an SSH client from your workstation.
3. Make sure the backup file that you intend to restore is located in `/usr/wave` directory. If it is not, copy, move, or link the file to that directory.
4. Change the directory to `/usr/wave/WAVEBackground`.
5. Run `./WAVEDBRestorer <backup-file>` where `<backup-file>` is the `backupSQLDump-`*`unique-date-timestamp`*`.wavedb` file that you want to restore.

What to do next

For usage information, enter **`WAVEDBRestorer`** *`-h | help | -?`*.

Regenerate IBM Wave Database Password

Use the **Regenerate IBM Wave Database Password** to change the password that is used by the Background Task Scheduler (BTS) to access the IBM Wave database.

The **Regenerate IBM Wave Database Password** action is available from the IBM Wave main menu. To access it, select **Administrative** > **IBM Wave Database Actions** > **Regenerate IBM Wave Database Password**. The password is saved as a hash file in the `/usr/wave/.databaseHashFile` directory.

Regenerate Encryption Keys

Use the **Regenerate Encryption Keys** action to regenerate the encryption keys that are used by IBM Wave to encrypt data at rest.

The **IBM Wave Regenerate Encryption Keys** function is available from the **IBM Wave Main Menu**. To access, select **Administrative** > **IBM Wave Database Actions** > **Regenerate Encryption Keys**.

Figure 45 on page 101 shows an example of the **WAVE Advanced Encryption Standard (AES) Regenerate Key Frame**. Click **Regenerate Keys** to replace the encryption key.

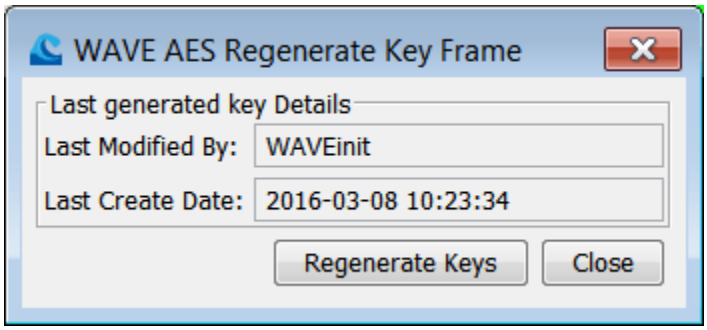


Figure 45. Regenerate Encryption Keys

Manage IBM Wave Users

Use the **Manage IBM Wave Users** to add, delete or change the scopes and permissions for IBM Wave Users.

To manage IBM Wave Users, select **Administrative** > **Manage IBM Wave Users** to open the **IBM Wave User Manager**.

For complete information about the **IBM Wave User Manager** including adding and changing IBM Wave users, see “Understanding user types and roles” on page 147 and “Overview of scopes and permissions” on page 148.

Manage IBM Wave User Profiles

Use the **IBM Wave User Profiles** to classify IBM Wave users and grant scopes and permissions based on one or more LDAP group associations.

The fields in the **IBM Wave User Profile Manager** tabs are similar to the **IBM Wave User Manager** table. The key difference is the ability to associate an IBM Wave user profile with one or more Lightweight Directory Access Protocol (LDAP) groups. The association is helpful when you are using the IBM Wave LDAP integration to associate LDAP groups to user profiles. For setting the LDAP/Active directory options, see “Enterprise Directory parameters” on page 127.

For more information about the **User Profile Manager with LDAP group**, see “LDAP group-based security” on page 144.

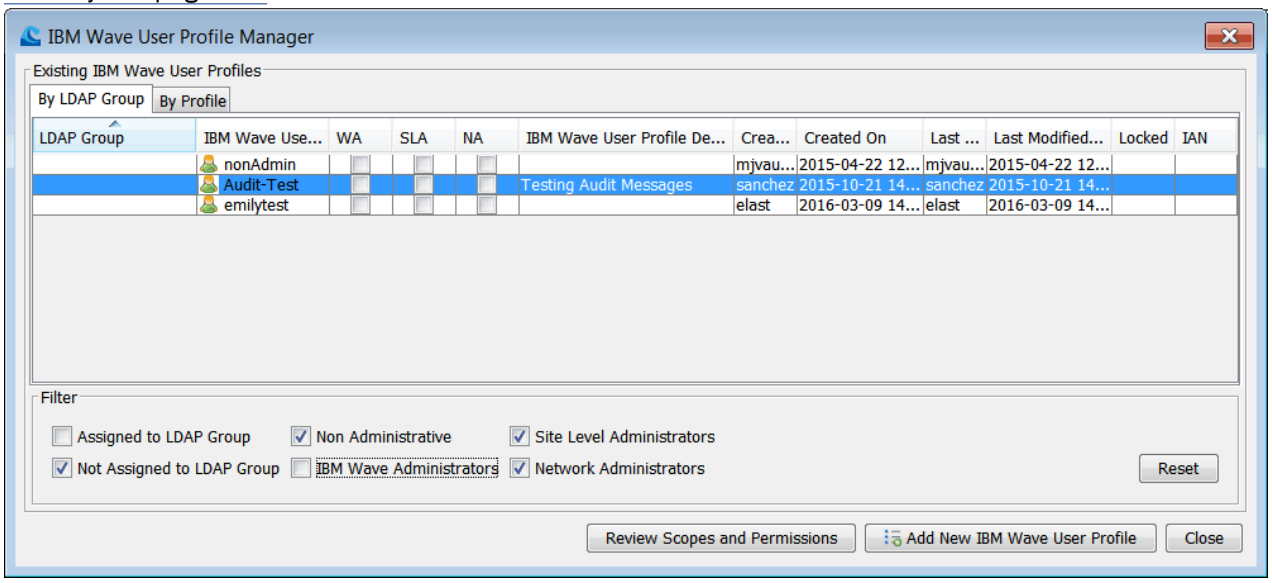


Figure 46. IBM Wave User Profile Manager By LDAP Group

For more information about using the **User Profile Manager By Profile**, see “Creating and updating IBM Wave user profiles” on page 155

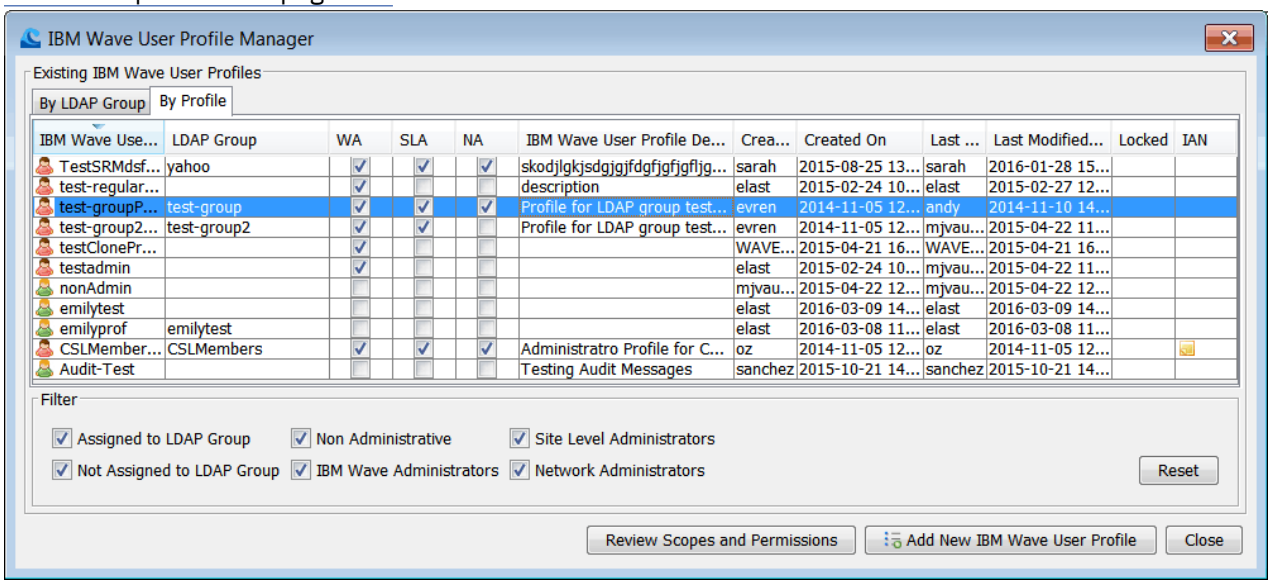


Figure 47. IBM Wave User Profile Manager By Profile

Project Manager

Using the **Project Manager**, you can define projects, and assign z/VM Guests and virtual servers to them. You can then view the guests that are grouped by project, or search and filter by project.

To access the **Project Manager**, click **Administrative** > **Project Manager**. The **Project Manager** opens as shown in Figure 48 on page 103.

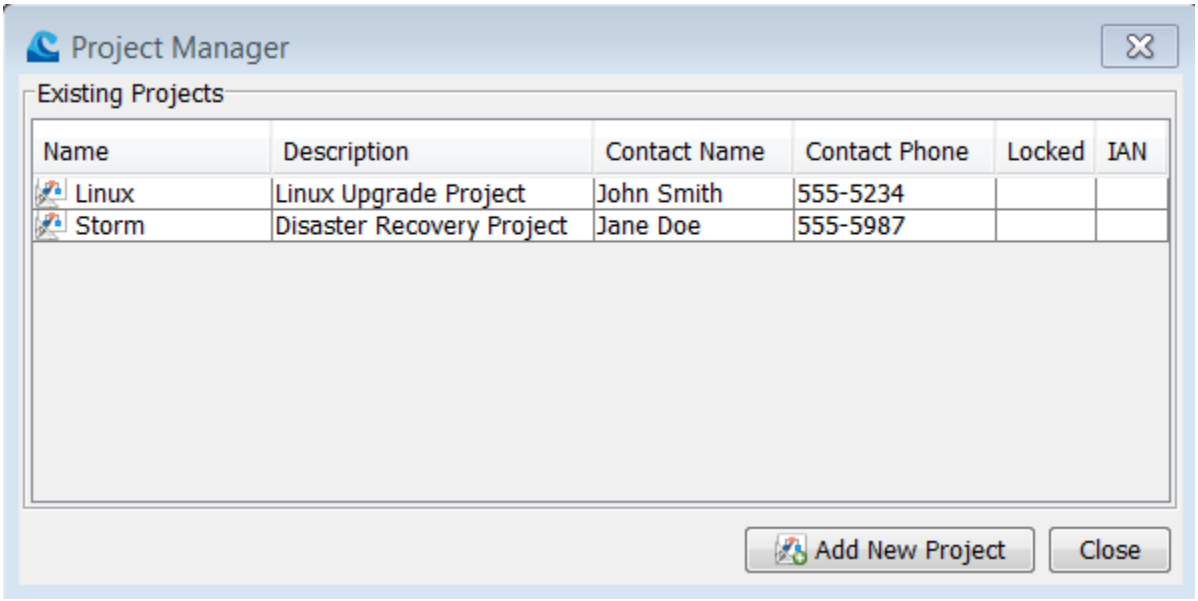


Figure 48. Project Manager

The following options are available:

- To add a Project, click **Add New Project**, and follow the instructions for "Add or Update a Project" on page 104.
- To update a Project, right-click on the project's row, and click **Update**. Next, follow the instructions for "Add or Update a Project" on page 104.
- To display an existing project, double-click the project's row.
- To add or edit an IAN attached to the Project, right-click on the project row and select **Read IAN** or **Update IAN**.
- To lock or unlock a project, right-click on the project row and click **Lock** or **Unlock**.
- To delete an existing project, right-click on the project row, and then click **Delete Project**.

  Note: You cannot delete a project if with z/VM Guests that are assigned to it. Depending on your scope and permissions, certain projects might appear not to contain z/VM Guests, but the delete option is still available. In this case, when you attempt to delete the project, an error message appears indicating that z/VM Guests are assigned to the project that are outside of your scope.

Important: When Automatic Guest Classification (AGC) is active and you assign guests to a project, all of the projects must use the "Bidirectional" rule. For more information, see "Automatic Guest Classification" on page 37.

For more information, see the following topics.

- "Automatic Guest Classification" on page 37
- "Add or Update a Project" on page 104
- "Metadata objects and entities" on page 17.

Add or Update a Project

To add a new project, select **Administrative** > **Project Manager** > **Add New Project**.

To update an existing project, in the **Project Manager**, right-click on the project row that you want to update, and then click **Update**.

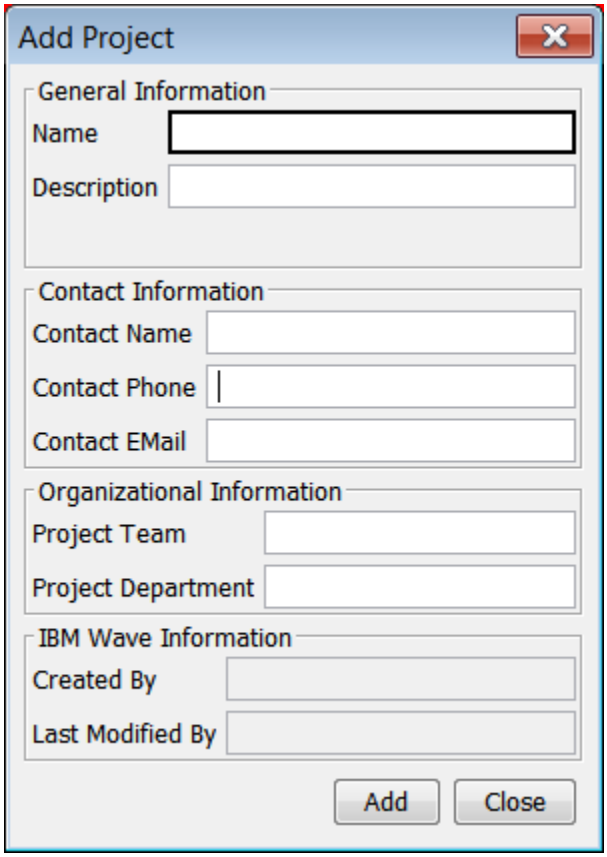


Figure 49. Add Project

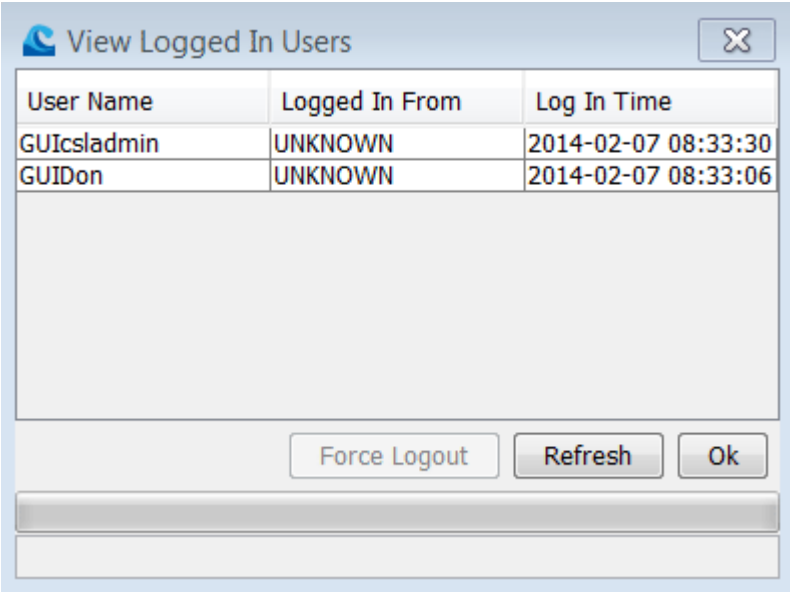
Complete the required field for the **Add Project**, or make the necessary changes when you **Update Project**. The following fields are available:

- **Name** - The *Name* for the project.

  Note: Names containing an asterisk ("*") or a semicolon (";") are not allowed.
- **Description** - Optional text that describes the project.
- **Contact Name** - The optional name of the contact for this project.
- **Contact Phone** - The optional phone number of the contact for this project.
- **Contact Email** - The optional email address of the contact for this project.
- **Team** - The optional name of the team in charge of this project.
- **Department** - The optional name of the department of the team.
- **Created By/Modified By** - Information only fields that describe who created and last updated the project.

View Logged in Users

This option is used to view all the logged in IBM Wave Users. The viewer allows the IBM Wave administrator to perform a "Force-logout" action on any logged-in IBM Wave Users.

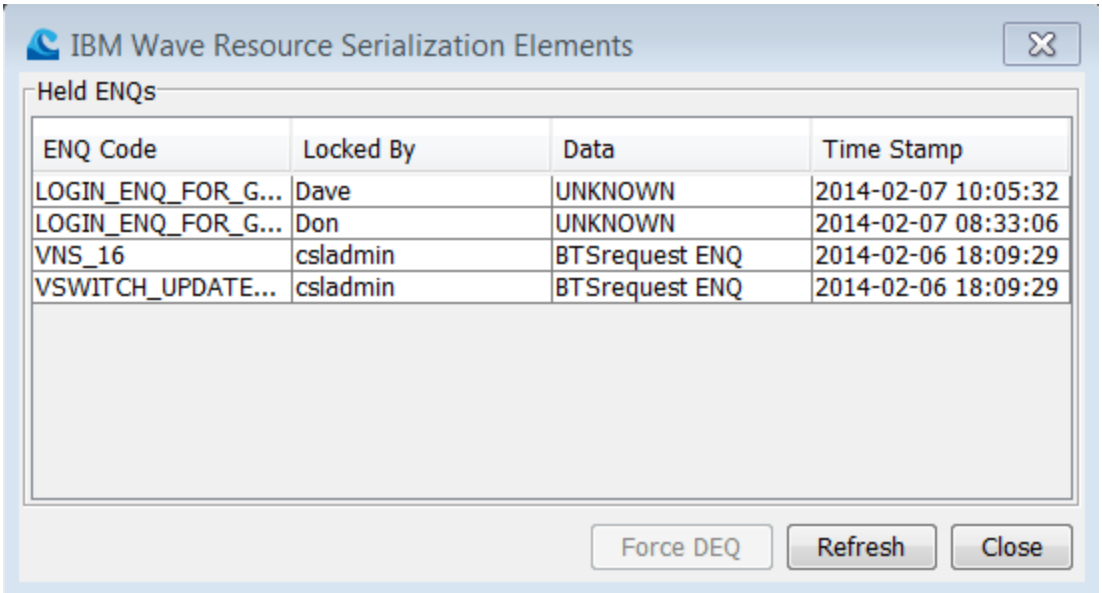


Note: This option actually displays all the LOGIN_ENQ_FOR WRS elements active in the system. IBM Wave creates a WRS element for each logged in IBM Wave User. If, for some reason, the IBM Wave User is disconnected from IBM Wave, without logging off normally, this WRS ENQ will remain in IBM Wave until one of the following occurs:

- The WRS Element is forcedly DEQ-ed by an IBM Wave administrator
- The IBM Wave User re logins into IBM Wave using the “Force” option and then logs off properly.

View WRS Elements

This Option is used to view and interact with the existing WRS ENQ elements in the IBM Wave System. The viewer allows the IBM Wave User to view the various WRS elements and ENQs active in the IBM Wave System and perform a force-DEQ action on these elements.



There are many types of WRS elements that are used by IBM Wave. Some of these elements disable certain functionality. Following is a list of the major types of WRS elements used by IBM Wave:

- LOGIN_ENQ_FOR_<IBM Wave User Name> - This element represents an IBM Wave User login session. Each IBM Wave User triggers the creation of this element. Once the IBM Wave User logs off from IBM Wave, the ENQ is deleted. If this ENQ is Force-DEQed, then the IBM Wave User assigned to it will be forcibly logged off from IBM Wave.

- <z/VM Virtual Server Name>_CLC_ENQ - This element represents a CLC instance for a certain IBM Wave User and a certain z/VM Virtual Server. This element is created whenever a CLC session is started by a certain IBM Wave User to a certain z/VM Virtual Server. DEQ-ing this element has no effect on the CLC session. For more information on CLC technology see “CLC technology” on page 24.

BTS Manager

The Background Task Scheduler (BTS) displays the current IBM Wave activity and scheduling definitions.

To open the **IBM Wave BTS Manager**, from the **IBM Wave Main Menu**, click **Administrative > BTS Manager**.

The **BTS Manager** displays the following information about BTS statistics such as:

- General Information
- Scheduling for tasks
- Internal BTS Request statistics
- BTS Log

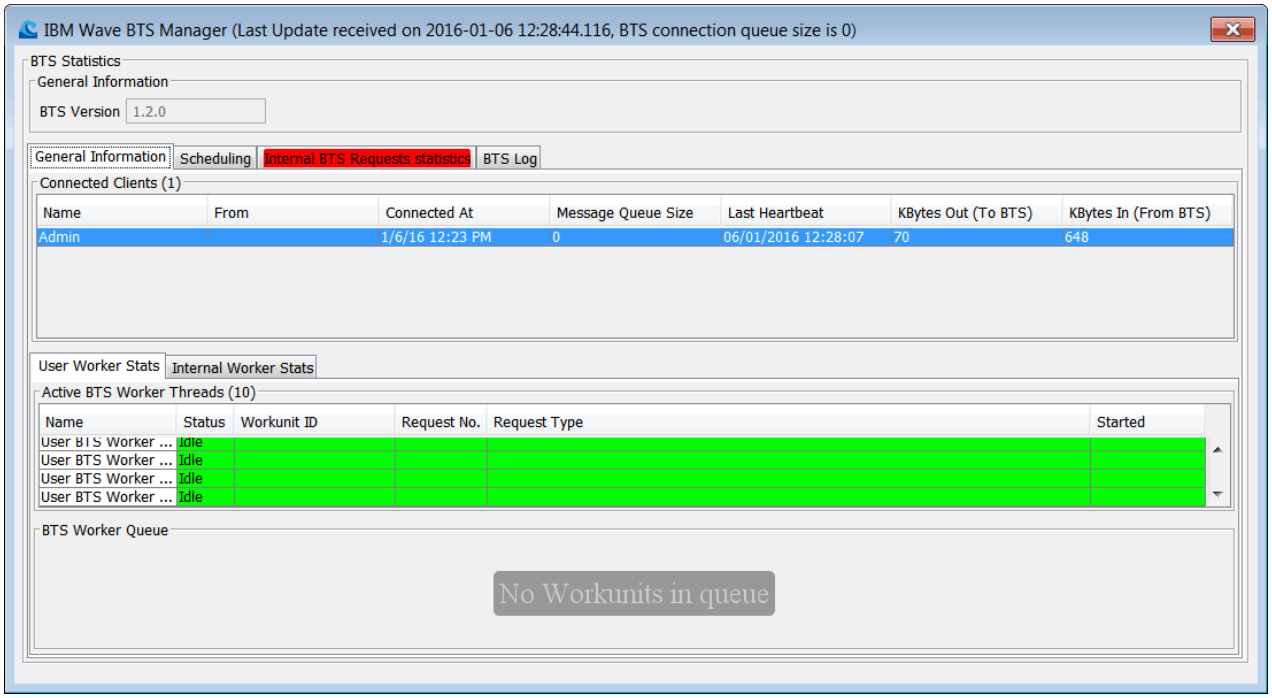


Figure 50. BTS Manager: General information

User Worker Stats tab: In the **Active BTS Worker Threads** table, right-click on an entry in the table to **Add BTS Worker Thread** or **Remove BTS Worker Thread** (as shown in Figure 51 on page 106).

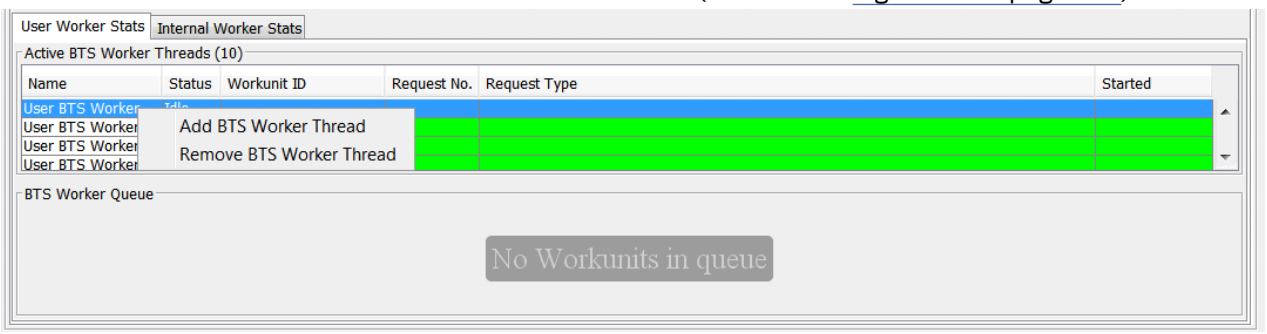


Figure 51. Add or remove a BTS worker thread

After you submit **Remove BTS Worker Thread** to the BTS, one of the idle BTS worker threads ends.

Notes:

- You cannot end a specific BTS worker thread.
- You cannot remove all worker threads; at least one thread must remain active.
- For information about setting the default number of **Active BTS Worker Threads**, see the **BTS tab** in “BTS parameters” on page 117.
- To customize the number of user worker threads, use the formula in “The Background Task Scheduler (BTS) server” on page 9.

BTS: Scheduling Tab

The **Scheduling** tab contains a table of all the scheduled BTS tasks.

- To activate or deactivate a scheduled entry, right-click the entry and select **Activate** or **Deactivate** from the menu.
- To run a specified task immediately, select **Run Now**.
- To change the scheduling interval of a task, click the "Schedule Parameter" cell for the corresponding task, and enter the new value.

For example, as shown in Figure 52 on page 107, you might decide to **Activate** the "Backup IBM Wave Database", and change the value from 24 hours to a value that suits your environment.

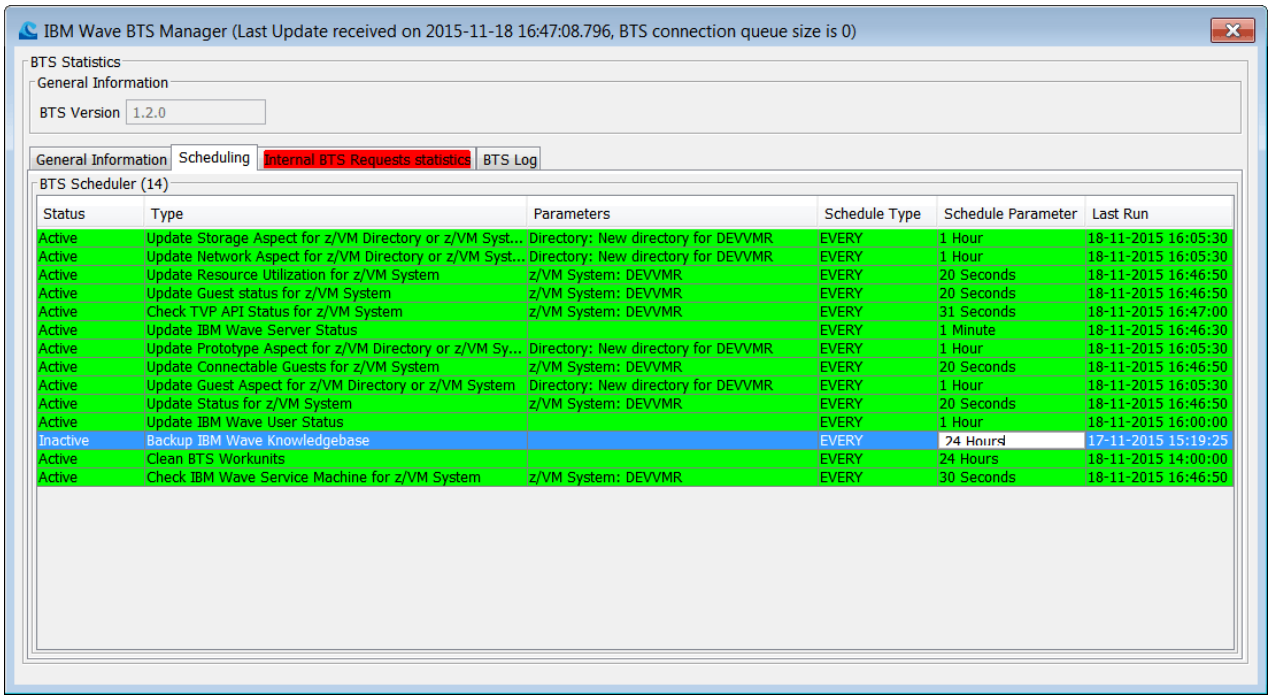


Figure 52. BTS Manager: Scheduling tab

BTS: Internal BTS Request statistics tab

The **Internal BTS Requests statistics** tab tracks periodic tasks that are run by the BTS. The table contains a list of all the periodic tasks that are run by the BTS and includes the run count, error count, and other statistics. If one or more tasks fail, the tab’s color changes to red, unless the tasks that are in error are marked "ignored".

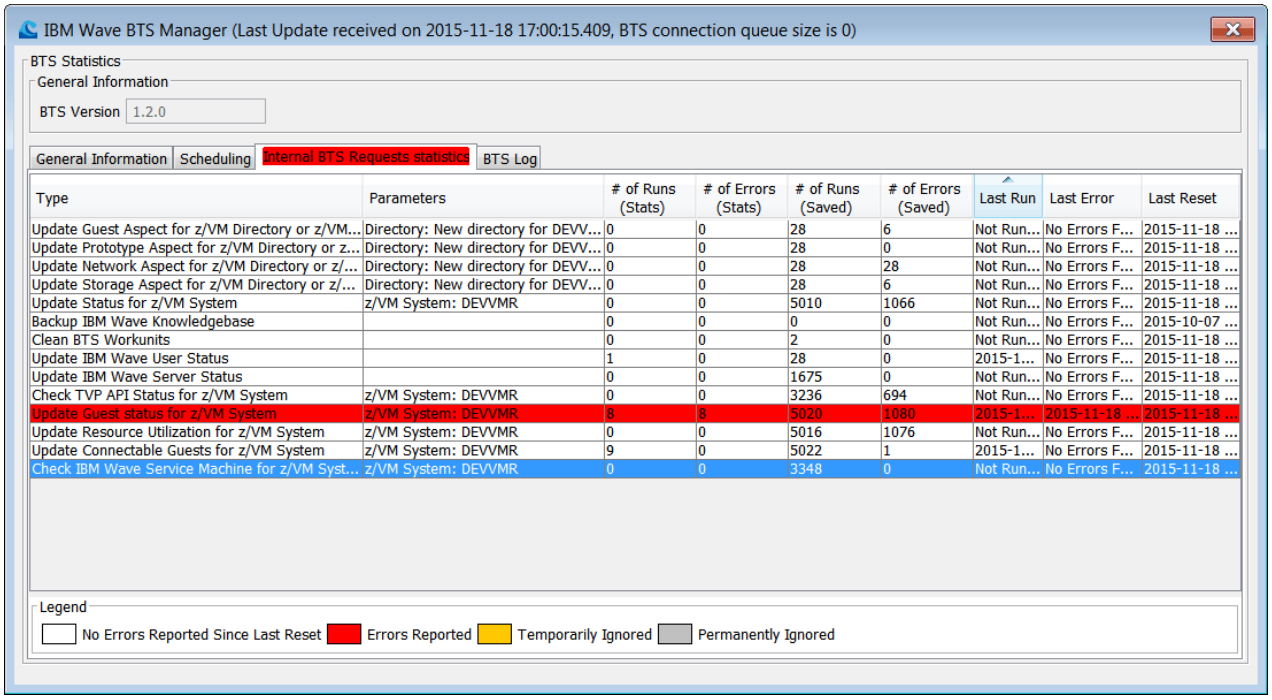


Figure 53. BTS Manager: Internal BTS Requests statistics

For the meaning of the color in the rows of the "Internal BTS Requests statistics" table, see Table 13 on page 108.

The following actions can be run against the table entries:

- **Reset Statistics** - Use this action to reset the run count, error count, and to put the current date and time in the last reset field.
- **Toggle "Ignore Until Next Error Occurs"** - Use this action to mark or unmark one or more entries as temporarily ignored. An entry that is temporarily ignored is ignored until the next time the task fails.
- **Toggle "Ignore Permanently"** - Use this action to mark or unmark one or more entries as permanently ignored. These entries are ignored regarding error counts and does not affect the color of the tab.

Table 13. Color legend for Internal BTS Requests

Color	Meaning
Red	Tasks that have an error count greater than zero, and are not permanently or temporarily ignored.
Gold	Tasks that are temporarily ignored.
Gray	Tasks that are permanently ignored.
White	Tasks that have an error count of zero.

Clean BTS work units

Use **Clean BTS Workunits** to delete BTS work units from the database.

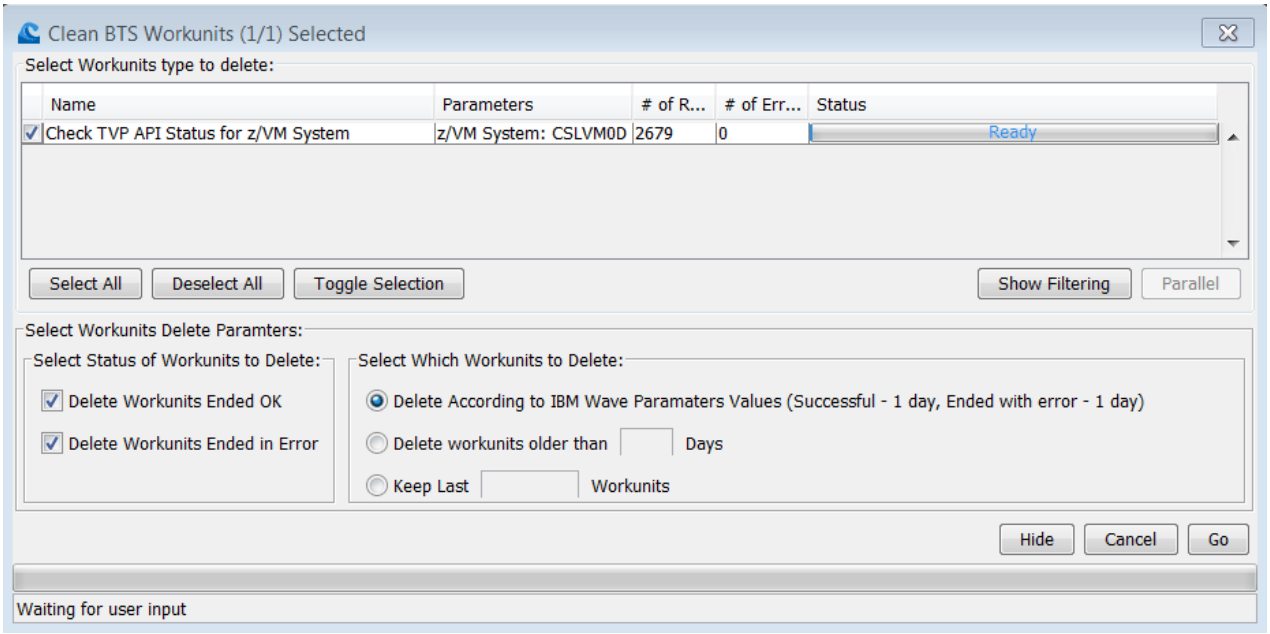


Figure 54. Clean BTS work units

To open the **Clean BTS Workunits** window (as shown in Figure 54 on page 109), right-click on a request in the **Internal BTS Requests statistics** tab.

The first table lists the type of BTS work unit for the delete action. By default the table contains all the BTS work unit types that are selected for the action.

Select Status of BTS Workunits to Delete:

- "Delete Workunits Ended OK" - Delete the BTS work units that ended without error.
- "Delete Workunits Ended in Error" - Delete the BTS work units that ended in error.

Select Which Workunits to Delete:

- "Delete According to IBM Wave Parameter Values" - Use to delete BTS work units based on the values that are specified in the IBM Wave parameters.
- "Delete work units older than *x* days" - Use to delete all BTS work units for the types that are selected whose end date is older than the value specified.
- "Keep Last *x* Workunits" - Use to keep the last *x* work units (in terms of end date) according to the value specified and deletes all others.

  Note: To delete all the BTS work units for the type that is selected, specify "0" for this option.

The BTS work units contains a delete request for each type selected. When you click **Go**, the BTS work unit is sent to the BTS.

Note: The Delete task deletes only BTS work units that were initiated by the internal BTS Scheduler. BTS work units that are issued by IBM Wave users are periodically cleaned according to the IBM Wave parameters by a different, internal BTS periodic task.

To retrieve all the internal BTS work units for the selected entry, double-click an entry. The displayed window can be used to view the COR outputs of the work units and requests, and behaves similar to the **BTS Workunit Viewer**.

Note: The BTS manager window automatically refreshes the view every five seconds.

Send Message

The **Send Message** option is used by an administrator to send a message to one or more z/VM Linux guests.

An IBM Wave administrator can use the **Send Message** option to send a message to a Linux guest. For example, when an administrator must notify users about systems that are being shut down.

To open the **Send Message** window, on the IBM Wave toolbar, click **Administrative > Send Message**. Enter the message text in the **Message** area. To send, click "**Go**".

- For more information, see “Single User Mode” on page 7 and https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_send_message1.html.
- For more information about the IBM Wave internal messaging mechanism, see “IBM Wave internal messaging mechanism” on page 7.

Broadcast Message to IBM Wave Users

An administrator can use the Broadcast Message to IBM Wave Users action to communicate with one or all IBM Wave users.

The **Broadcast Message** option is helpful when an administrator wants to notify users about a system event. For example, an administrator must enter single user mode to shut down a z/VM system. Several people are using IBM Wave, so the administrator broadcasts a message that requests all users log off IBM Wave by 9:00 PM.

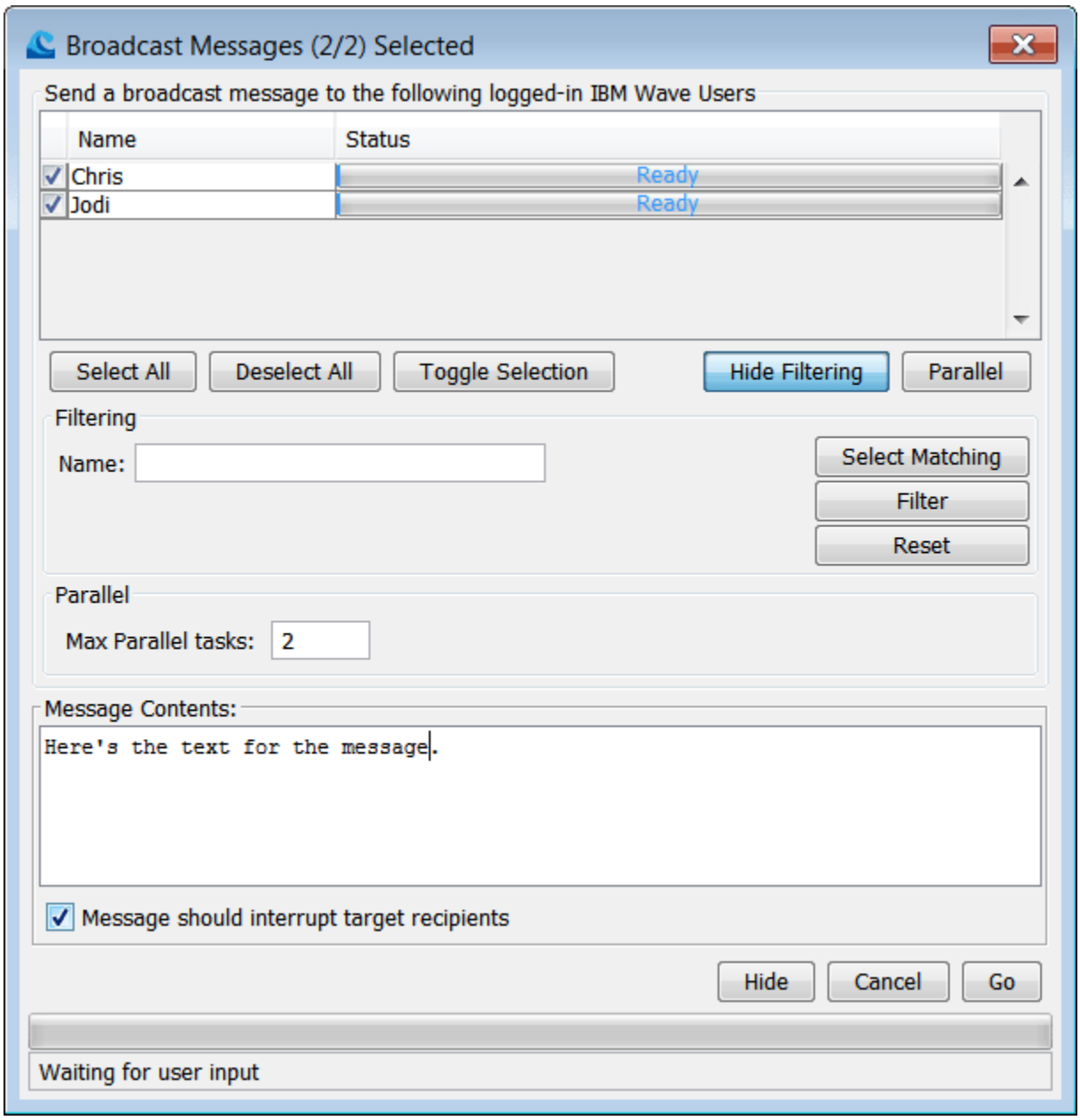


Figure 55. Broadcast message window

From the **IBM Wave main menu**, click **Administrative** > **Broadcast Message to IBM Wave users** to open the **Broadcast Message** window (as shown in Figure 55 on page 111).

Enter the message text in the **Message Contents** pane. When you are ready to broadcast, press "**Go**." IBM Wave displays a typical notification to tell the users that a new message from the administrator is waiting.

Occasionally, it might be necessary to interrupt a user. Select the "**Message should interrupt targeted recipients**" check box. IBM Wave sends an urgent message that must be manually cleared before the user can resume work on another IBM Wave task.

The **Broadcast Message** window contains the following options:

Toggle Selection
Use to switch between selected users.

Select or Deselect All
Use to select or clear the complete list of users.

Show Filtering
Use to display the Filtering pane. The Filtering pane uses the name column as a filter, and accepts an asterisk (*) as a wildcard. (For example, *C**).

Hide Filtering
Use to conceal the Filtering pane.

Parallel
Use to broadcast the message on multiple z/VM systems in parallel.

For more information about the messaging mechanism, see the topic about "IBM Wave internal messaging mechanism" on page 7.

Recycle Service Machines

An administrator can use the **Recycle Service Machines** action to restart the service machines on the selected z/VM system.

The **Recycle Service Machines** action recycles the IBM Wave service machines on the z/VM system that is currently selected. The action warns the IBM Wave user if there are any users of the service machines.

To recycle the IBM Wave service machines, from the IBM Wave main menu, select **Administrative** > **Recycle Service Machines**.

Recycle API servers

An administrator can use the **Recycle API servers** option to restart the API servers on the selected z/VM System.

The **Recycle API servers** action uses the IBM Wave service machines to recycle the API servers on the currently selected z/VM System. The service machines are used to implement this feature. If the service machines are not operational, the **Recycle API servers** menu item is disabled.

To restart the API servers, click **Administrative > Recycle API servers** from the main menu.

Manage Parameters

An administrator can use **Manage Parameters** to control the global parameters for IBM Wave.

Use the **Manage Parameters** option to modify the IBM Wave parameters. For complete information, see Chapter 5, "System customization," on page 113.

Chapter 5. System customization

IBM Wave provides parameters for you to customize your experience with the application and interface.

IBM Wave parameters

Use the IBM Wave parameters to customize the IBM Wave environment.

To customize the IBM Wave parameters, from the IBM Wave main menu, click **Administrative** > **Manage Parameters**.

The **Manage Parameters** window is composed of tabs that represent aspects of the IBM Wave application interface. The parameters are global and affect all IBM Wave users of the specified IBM Wave server. When you are done making updates, press **Update** to store the parameter changes in the IBM Wave database.

The following IBM Wave parameters are available:

Thresholds and Defaults

Use the **Thresholds and Defaults** parameters to set the defaults for dynamic CPU and dynamic memory values and modify the z/VM guest's disk space, spool, page, Linux, and virtual to real thresholds.

To access the **Thresholds and Defaults** parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Thresholds and Defaults**.

The first tab that you see is named **Thresholds and Defaults** (as shown in Figure 56 on page 114).

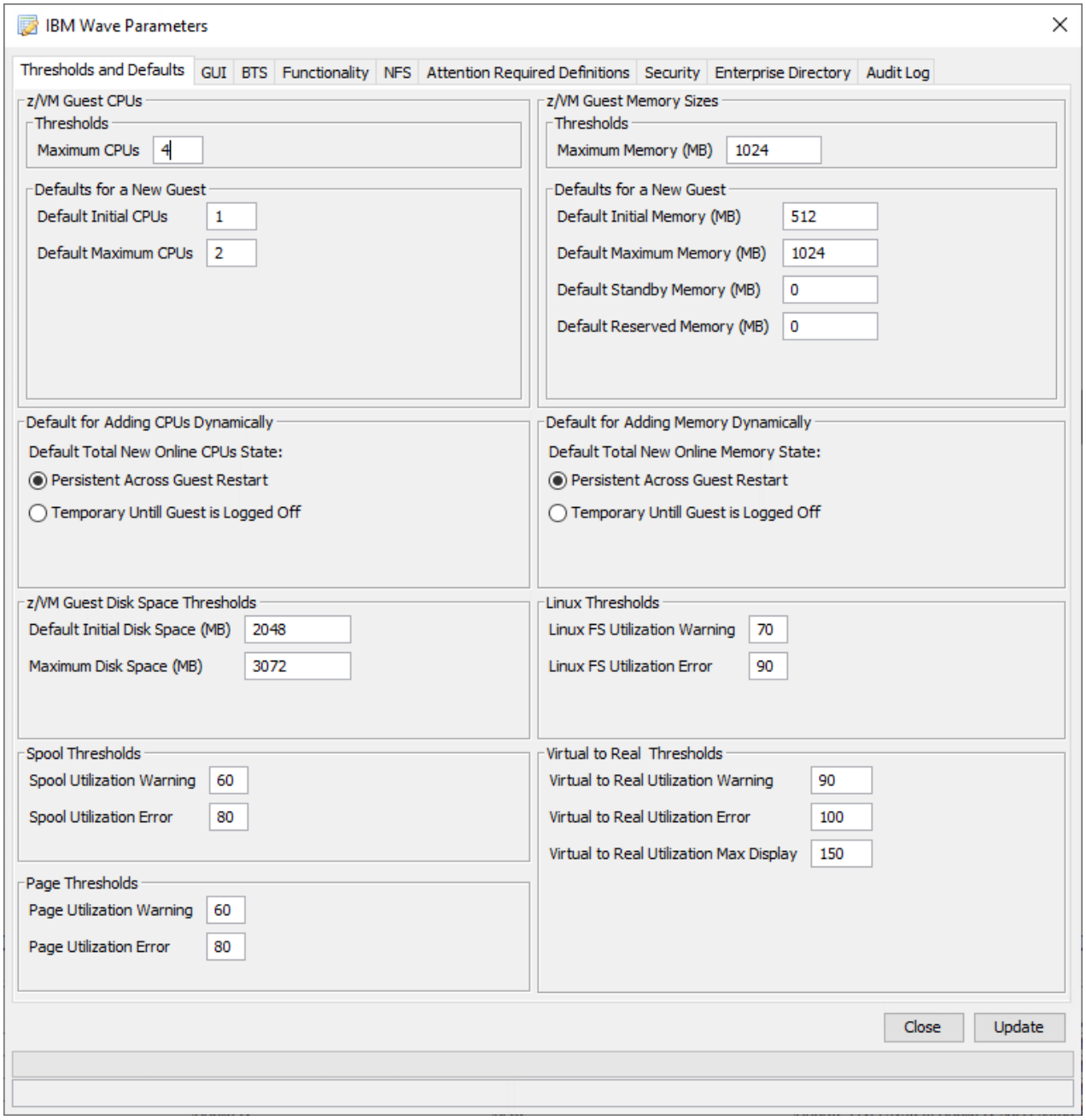


Figure 56. Thresholds and Defaults tab

z/VM Guest CPUs

Thresholds

Maximum CPUs - The maximum number of CPUs that can be specified.

Defaults for a New Guest

Default Initial CPUs - The initial number of CPUs that can be specified for each new z/VM Guest.

Default Maximum CPUs - The maximum number of CPUs that can be specified by default for each new z/VM Guest.

Defaults for Adding CPUs Dynamically

Default Total New Online CPU State

Persistent Across Guest Restart - Indicates that the default CPU setting is persistently across guest restart. The **Persistent Across Guest Restart** is disabled if a user does not have "Update" authority to the guest.

Temporary Until Guest is Logged Off- Indicates that the default CPU setting is saved temporarily until the next time the guest is restarted.

z/VM Guest Disk Space Thresholds

Default Initial Disk Space (MB) - The default storage space that is initially allowed for each z/VM Guest.

Maximum Initial Disk Space (MB) - The maximum storage that is allowed for each z/VM Guest.

Spool Thresholds

Spool Utilization Warning - A percentage (0 - 99). The value is checked against the current usage level of the Spool. If the value exceeds the specified percentage, it is indicated in the **Spool Utilization Dial** chart.

Spool Utilization Error - A percentage (0 - 99). The value is checked against the current usage level of the Spool. If the value exceeds the specified percentage, it is indicated in the **Spool Utilization Dial** chart. And the z/VM System is marked with an "In Error" flag.

Note: The error value must exceed the value that is specified for the **Spool Utilization Warning**.

Page Thresholds

Page Utilization Warning - A percentage (0 - 99). The value is checked against the current usage level of the Page disks. If the value exceeds the specified percentage, it is indicated in the **Spool Utilization Dial** chart.

Page Utilization Error - A percentage (0 - 99). The value is checked against the current usage level of the Page disks. If the value exceeds the specified percentage, it is indicated in the **Spool Utilization Dial** chart. And the z/VM System is marked with an "In Error" flag.

Note: The error value must exceed the value that is specified for the **Page Utilization Warning**.

Defaults for a New Guest

Default Initial Memory (MB)
The default initial amount of memory, in MB, for each new z/VM Guest.

Default Maximum Memory (MB)
The default maximum amount of memory, in MB, for each new z/VM Guest.

z/VM Guest Memory Sizes
Thresholds

Maximum Memory (MB) - The threshold value (maximum) for maximum memory, in MB, that can be specified.

Defaults for New Guest

Default Initial Memory (MB) - The default amount of initial memory, in MB, that can be assigned to each new z/VM guest.

Default Maximum Memory (MB) - The default amount of maximum memory, in MB, that can be assigned to each new z/VM guest.

Default Standby Memory (MB) - The default amount of standby memory, in MB, that can be assigned to each new z/VM guest.

Default Reserved Memory (MB) - The default amount of reserved memory, in MB, that can be assigned to each new z/VM guest.

Important: Dynamic memory reconfiguration is supported only when the initial memory size for the guest is an exact multiple of the memory block size. To understand the calculation of memory block size, see the following topics:

- **Defining storage:**
  - CP DEFINE STORAGE command (z/VM 6.4)
  - CP DEFINE STORAGE command (z/VM 7.1)
  - CP DEFINE STORAGE command (z/VM 7.2)
- **Planning for Linux virtual servers:**

- Memory and CPU requirements (z/VM 6.4)
- Memory and CPU requirements (z/VM 7.1)
- Memory and CPU requirements (z/VM 7.2)

Note: For preexisting Linux guests, which never had values for **Standby Memory** or **Reserved Memory** defined, the values are automatically set to zero.

Defaults for Adding Memory Dynamically
Default Total New Online Memory State:

Persistent Across Guest Restart - Indicates that the default memory state is persistently across guest restart. When a user does not have **Update** authority to the guest, the **Persistent Across Guest Restart** state is disabled.

Temporary Until Guest is Logged Off - Indicates that the default memory state is temporary until the guest logs off. When a user does not have **Update** authority for the guest, the **Temporary Until Guest is Logged Off** is the default state.

Linux Thresholds

Linux FS Utilization Warning - A percentage (1-100). The value is checked when the Linux file system information (in the **Display information** or **Manage Storage** actions) is displayed. File system whose usage exceeds this amount is colored in Orange.

Linux FS Utilization Error - A percentage (1-100). The value is checked when the Linux file system information (in the **Display information** or **Manage Storage** actions) display. File system whose usage exceeds this amount is colored in red. The value must exceed the value that is specified in the **Linux FS Utilization Warning**.

Note: IBM Wave also monitors the WAVESRV server file system usage. If the usage level of any of the file systems in the server are equal to or exceed 95%, IBM Wave issues a warning message when you open IBM Wave.

Virtual to Real Thresholds

Virtual to Real Utilization Warning - A percentage (0 - 99). The value is checked against the current usage level of the virtual to real storage usage. If the value exceeds the specified percentage, it is indicated in the **Virtual to Real Utilization Dial** chart.

Virtual to Real Utilization Error - A percentage (0 - 99). The value is checked against the current usage level of the Virtual to Real Storage. If the value exceeds the specified percentage, it is indicated in the **Virtual to Real Utilization Dial** chart and the z/VM system is marked with an `In Error` flag.

Virtual to Real Utilization Maximum Display - Indicates the percentage of the maximum that the **Virtual to Real Utilization Dial** chart shows. For example, if you enter 150, it allows for the display of the maximum value (99) plus an extra 51 percent (providing that the maximum is exceeded).

GUI parameters

To access the graphical user interface (GUI) parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **GUI**.

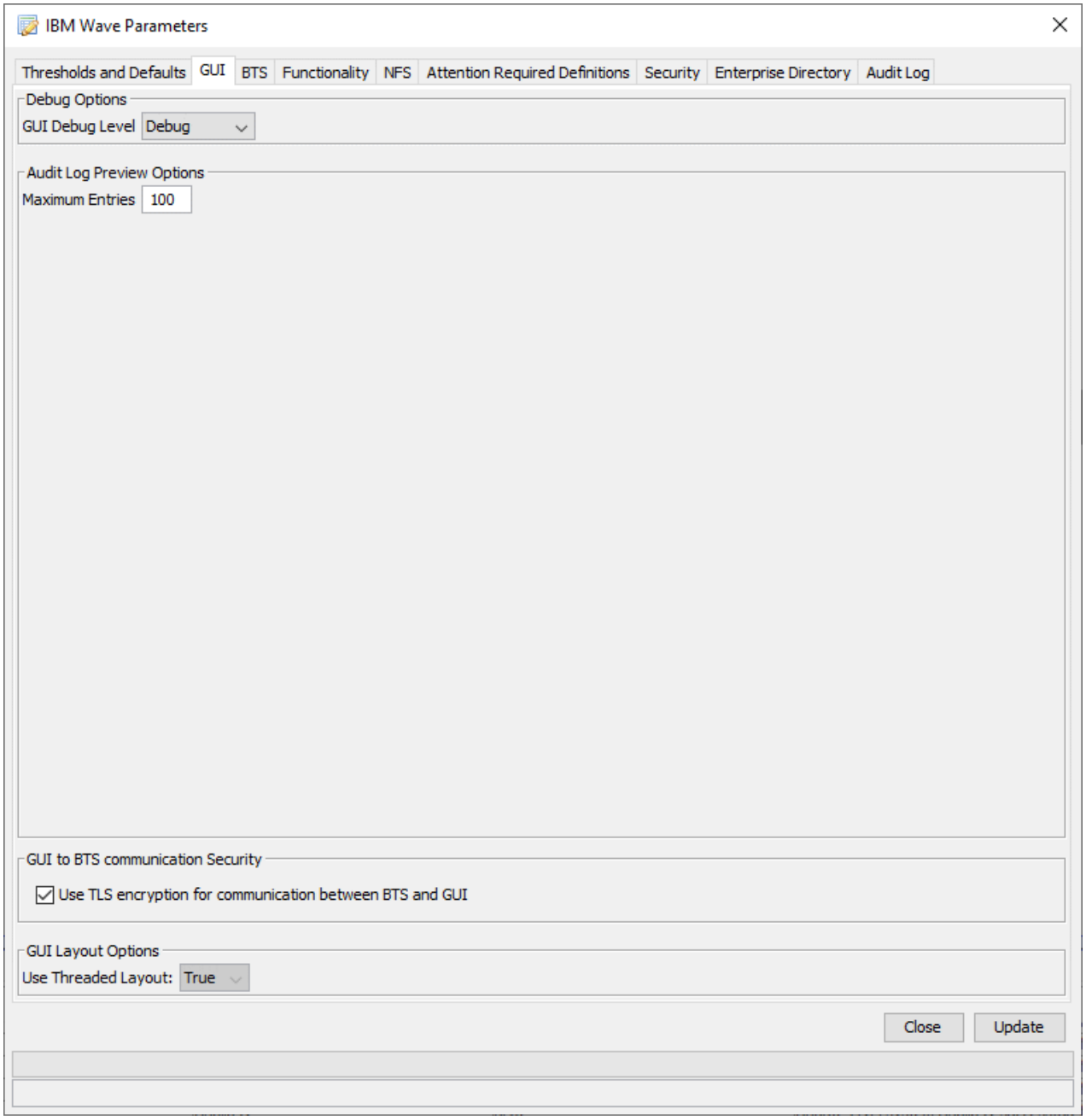


Figure 57. IBM Wave parameters: GUI tab

- **GUI Debug Level** - The debugging level of the GUI client. The debug option is relevant when you select **User Tasks > Trace GUI** from the IBM Wave main menu.
- **Audit Log Preview Options** - The maximum number of log entries that appear in the **Audit Log Preview** tab (in the **General Status Viewer**). The default is 100. You can optionally specify 10 - 100.
- **GUI to BTS communication Security** - The option to specify Transport Layer Security (TLS) encryption between the BTS and the GUI. A checked box is the default. If the box is not checked, the connection is unencrypted and is not secure.
- **GUI Layout Options** - The threaded layout is the default.

BTS parameters

To access the Background Task Scheduler (BTS) parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **BTS**.

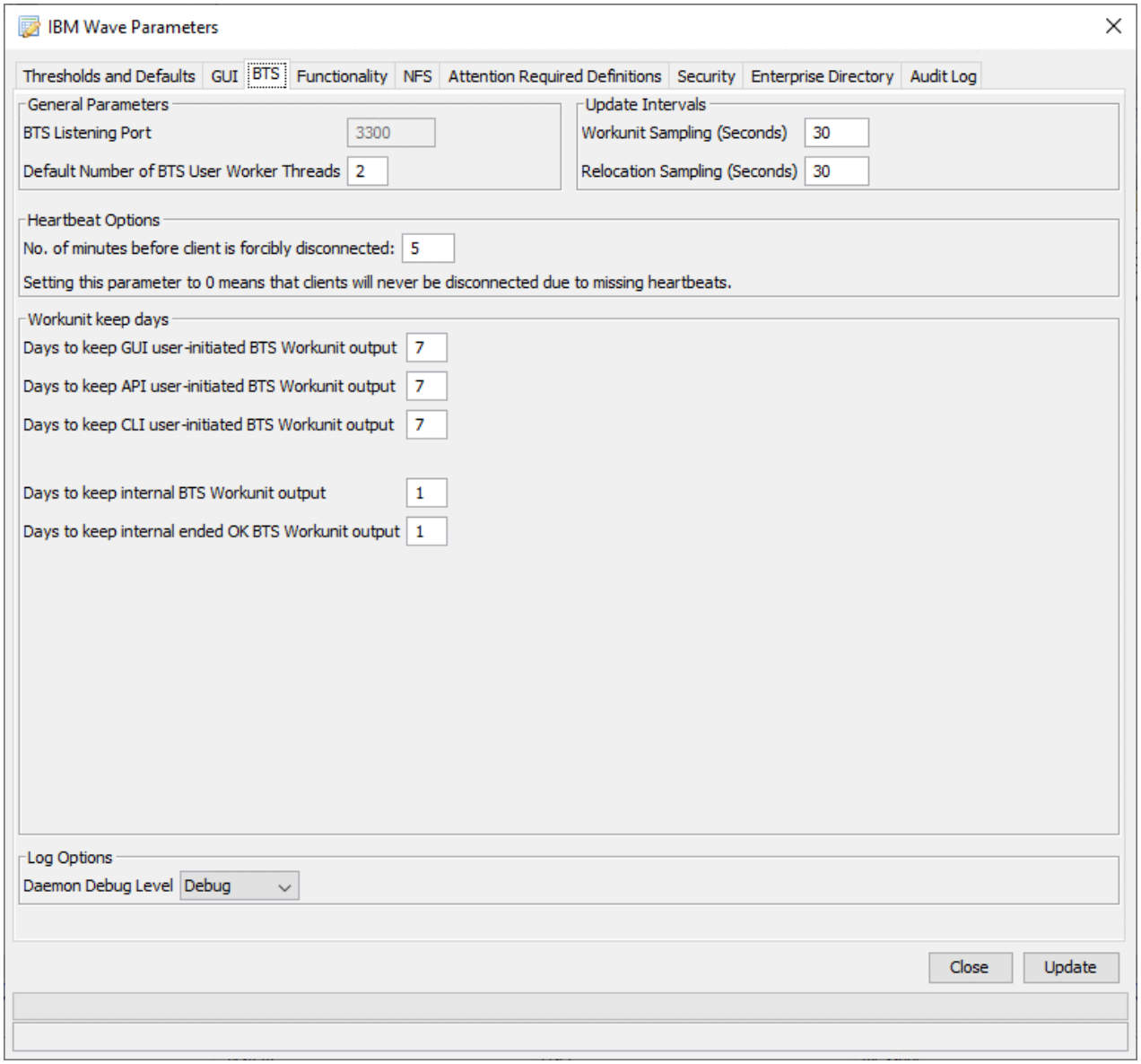


Figure 58. IBM Wave Parameters: BTS

The BTS tab contains the following options:

General Parameters

BTS Listening Port

The port on which the BTS is listening for requests. Only a Wave server Linux administrator can change the port number. To change the port number *during installation*, use the procedure in step “8” on page 64. To change the port number *when updating Wave* (to a new fix pack, for example), use the procedure in “Upgrading IBM Wave for z/VM” on page 68.

Default Number of BTS User Worker Threads

The default number of BTS User Worker Threads that are active. The default number matches the number that is identified in the BTS Manager's **Active BTS Worker Threads** table.

Update Intervals

Workunit Sampling (Seconds)

The interval (in seconds) that is used to sample the Directory Manager-driven work units when they exist in the system.

Relocation Sampling (Seconds)
The interval (in seconds) that is used to sample relocation processes (when relocation processing exists in the system).

Heartbeat Options

No. of minutes before client is forcibly disconnected
The number of minutes before a BTS client is forcibly disconnected.

Work Unit Keep Days

Days to keep GUI user-initiated BTS Workunit output
This value indicates the number of days that the BTS keeps GUI user-initiated BTS work units. The periodic cleaning task deletes all GUI user-initiated BTS work units whose end time is older than the specified number of days.

Days to keep API user-initiated BTS Workunit output
This value indicates the number of days that the BTS keeps API user-initiated BTS work units. The periodic cleaning task deletes all API user-initiated BTS work units whose end time is older than the specified number of days.

Days to keep CLI user-initiated BTS Workunit output
This value indicates the number of days that the BTS keeps CLI user-initiated BTS work units. The periodic cleaning task deletes all CLI user-initiated BTS work units whose end time is older than the specified number of days.

Days to keep internal BTS Workunit output
This value indicates the number of days that the BTS keeps internal BTS work units. The periodic cleaning task deletes all internal BTS work units whose end time is older than the specified number of days. For internal BTS work units that ended successfully, there's a separate parameter. If the separate parameter's value is different, the periodic cleaning task takes that parameter's value when it is deleting the BTS work units that ended successfully.

Days to keep ended OK internal BTS Workunit output
This value indicates the number of days that the BTS keeps the internal BTS work units that ended successfully. The periodic cleaning task deletes all internal BTS work units that ended successfully and whose end time is older than the specified number of days.

Log Options

Daemon Debug Level
The debug level for the BTS.

Functionality parameters

The functionality parameters control such configuration options as activation levels, SSH options, communication, and API timeout settings.

To access the functionality parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Functionality**.

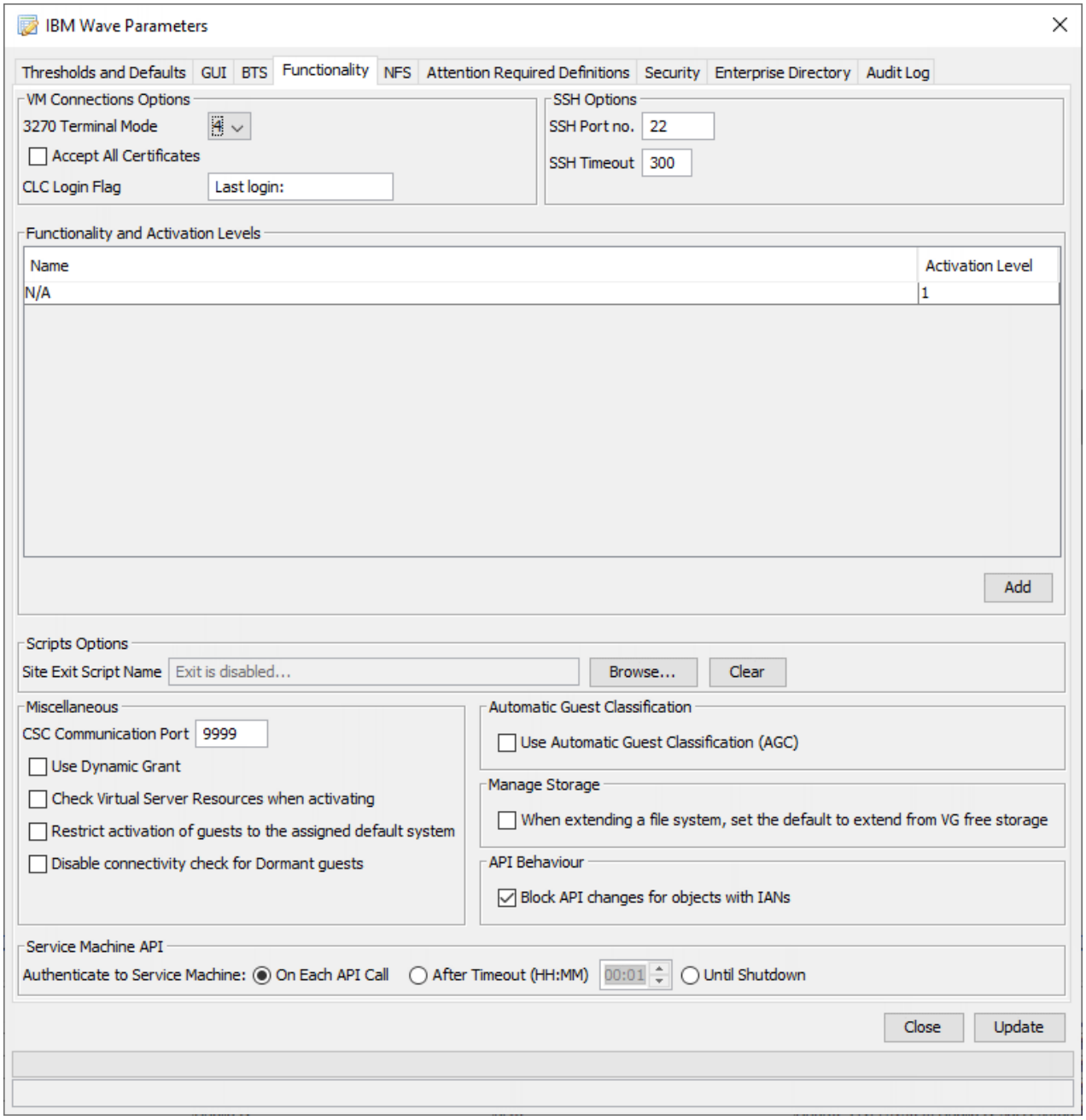


Figure 59. IBM Wave parameters Functionality tab

z/VM Connection Options

3270 Terminal Mode
The 3270 terminal mode for access.

Accept All Certificates
Controls the amount of certificate validation that IBM Wave performs when creating encrypted 3270 or CLC sessions.

When the box is unchecked (the more secure option), the server's certificate must be valid in order to create a secure 3270 or CLC session; if certificate validation fails, no connection is established. When you first install IBM Wave, the default is that the z/VM system's server certificate must be valid in order to establish an encrypted connection.

When the box is checked, any server's certificate, including a self-signed or invalid certificate, can be used to create a secure 3270 or CLC session.

This option applies to all z/VM systems with guests to which you might connect using 3270 or CLC.

CLC Login Flag
The text string that symbolizes a successful login to a Linux server by using CLC. This parameter must be set to the first prompt line that appears at login for the Linux server.

SSH Options

SSH Port
The port on which the SSH servers on the Linux z/VM guests are listening.

SSH Timeout
The timeout value is used by the Background Task Scheduler (BTS) to check when the z/VM guests are connectable. The default value is 300 ms.

Functionality and Activation Levels - The **Functionality and Activation Levels** table contains the defined Functionality Names and the Activation Levels that are assigned to each function. To define a new Functionality Name, click **Add**, and then complete the Functionality Name and the Activation Level fields. To remove an existing Functionality Type entry, right-click the row and select "Delete". To update or change a functionality name, or change the assigned activation level, right-click, and select "Update". For more information, see “Functionality and Activation Levels and Activation Done signaling” on page 21.

Script Options

Script Exit Script Name
Indicates the default exit script that runs before the IBM Wave Script executor runs the user-specified script. Do so to make environmental changes before IBM Wave runs a generic script. Click **Browse** to choose a script from the available IBM Wave scripts.

Miscellaneous

CSC Communication Port
The port the CSC service machine uses for minidisk streaming.

Use Dynamic Grant
When checked, IBM Wave can issue dynamic GRANT commands to VSwitches upon activation of z/VM guests that are connected to those VSwitches. For more information about the usage of dynamic and static GRANT processing, see “Dynamic and static GRANT processing” on page 29.

Check Virtual Server Resources when activating
When checked, IBM Wave automatically runs several checks before z/VM Guest Activation. IBM Wave can activate the z/VM guest only when all the checks return normally. For more information about the checks for z/VM Guest activation, see “Resource verification before activation” on page 21.

Restrict activation of guests to the assigned default system
When checked, only a user who has the system level administrator (SLA) role can activate a guest that is not on its default system. For example, if DEVVMR is the default z/VM system for the guests, anyone with *Activate* permission can activate the guests. If **Restrict activation of guests to the assigned default system** is selected but DEVVMR is not the default system, only the SLA can activate the guests.

Disable connectivity check for dormant guests
As part of its background tasks, IBM Wave checks the connectivity of Linux guests. When this option is selected, IBM Wave detects and marks guests with low CPU consumption as dormants. Such a guest is not probed by the background task and its connectivity status does not change until the guest becomes active.

Automatic Guest Classification (AGC)

Use Automatic Guest Classification (AGC)
When AGC is selected, IBM Wave provides a means to tightly couple IBM Wave metadata elements with z/VM guests' directory entries. For more information about Automatic Guest Classification, see “Automatic Guest Classification” on page 37.

Manage Storage

When extending a file system, set the default to extend from VG free storage
When checked, IBM Wave supports extending storage from both new or preexisting free storage that is defined in a user's volume group (VG). For information about this option, see "Storage management" on page 27.

API Behavior

Block API changes for objects with IANs
When checked, an IBM Wave API call will receive a 409 response when run against objects (such as z/VM guests or z/VM systems) that have IANs attached to them. A checked box is the default.

Service Machine API - The timeout value that is specified to validate the connection between the BTS and the service machine.

On Each API Call
Authenticate on each API call.

After Timeout
Authenticate on the next API call after a specified timeout period of 1 minute to 1440 minutes (24 hours).

Until Shutdown
Remain authenticated until the BTS or the service machine is shut down.

NFS parameters

This topic shows how to update the Network File System (NFS) parameters.

To access the Network File System (NFS) parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **NFS**.

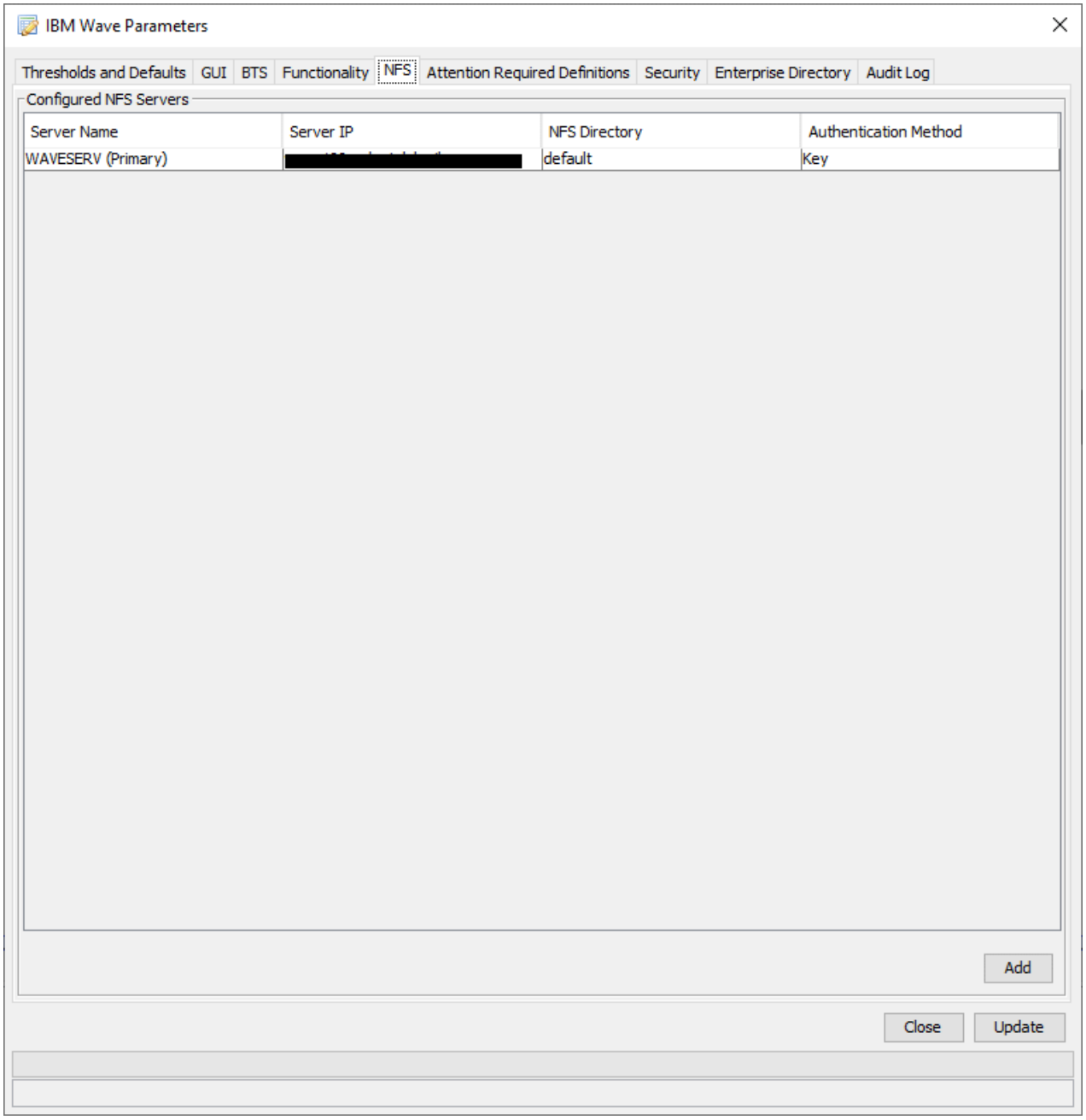


Figure 60. NFS parameters

The **Configured NFS Servers** table contains all defined NFS parameters for servers with the Name, IP address, NFS export directory, and authentication method.

- To add an NFS server, click **Add**.
- To update an entry, right-click in the table and click **Update**.
- To delete an entry, right-click in the table and click **Delete**.

  Note: You cannot delete an NFS server entry that is used as the default server for one or more z/VM Systems.

Click **Update** to save changes to the **Configured NFS Server** table. When a new NFS server is added, a BTS Script Sync request is initiated to sync all the existing scripts to the new NFS server. If an NFS server is deleted, scripts are not removed (because IBM Wave no longer has the login credentials to delete).

Note: After you press **Update**, the credentials are not saved.

Click **Add** to define a new NFS server. Click **Update** to update an existing server. The **Add New NFS Server** or **Update NFS Server** window appears.

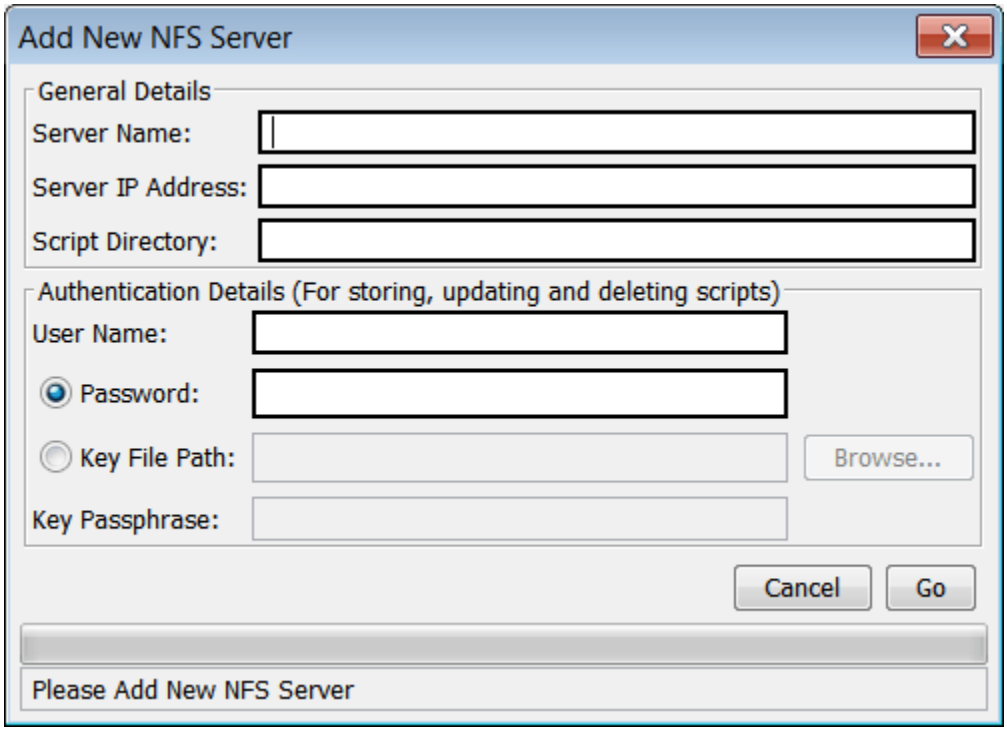


Figure 61. Add New NFS Server

- **Server Name** - A user-defined name for the NFS Server.
- **Server IP address** - The IPv4 address of the NFS Server.
- **Script Directory** - The directory on the NFS Server that is exported by the NFS Server for NFS Mounting by z/VM Guests.

Authentication Details

Authentication Details (For storing, updating, and deleting scripts)

Use one of the following methods to authentication to the NFS server:

- Specify a **User Name** and **Password**
- Click **Browse** to locate the **Key File Path** and enter the **Key Passphrase**.

When you are finished updating or adding a server, authenticate, and then click **Go**.

Notes:

1. The credentials are used to access the server through Secure FTP to sync the scripts. The credentials are not used for the NFS mount.
2. The credentials are encrypted and decrypted each time the NFS server is used. If you try to edit the NFS server, you must type the credentials again.

Attention Required Definitions

To access the **Attention Required Definitions** from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Attention Required Definitions**.

Use the **Attention Required Definitions** to customize the way that error events are handled by IBM Wave. You can take the following

- Customize the default severity on an event basis. To customize the **Default Severity** column, click the cell and change the value.

- Specify that certain events be ignored. To ignore an event, select the **Ignored** check box next to the event you want to ignore.

  Note: Ignoring an attention required event causes the event not to be displayed for the object. The icon for the object is also not displayed with any warning indicators.

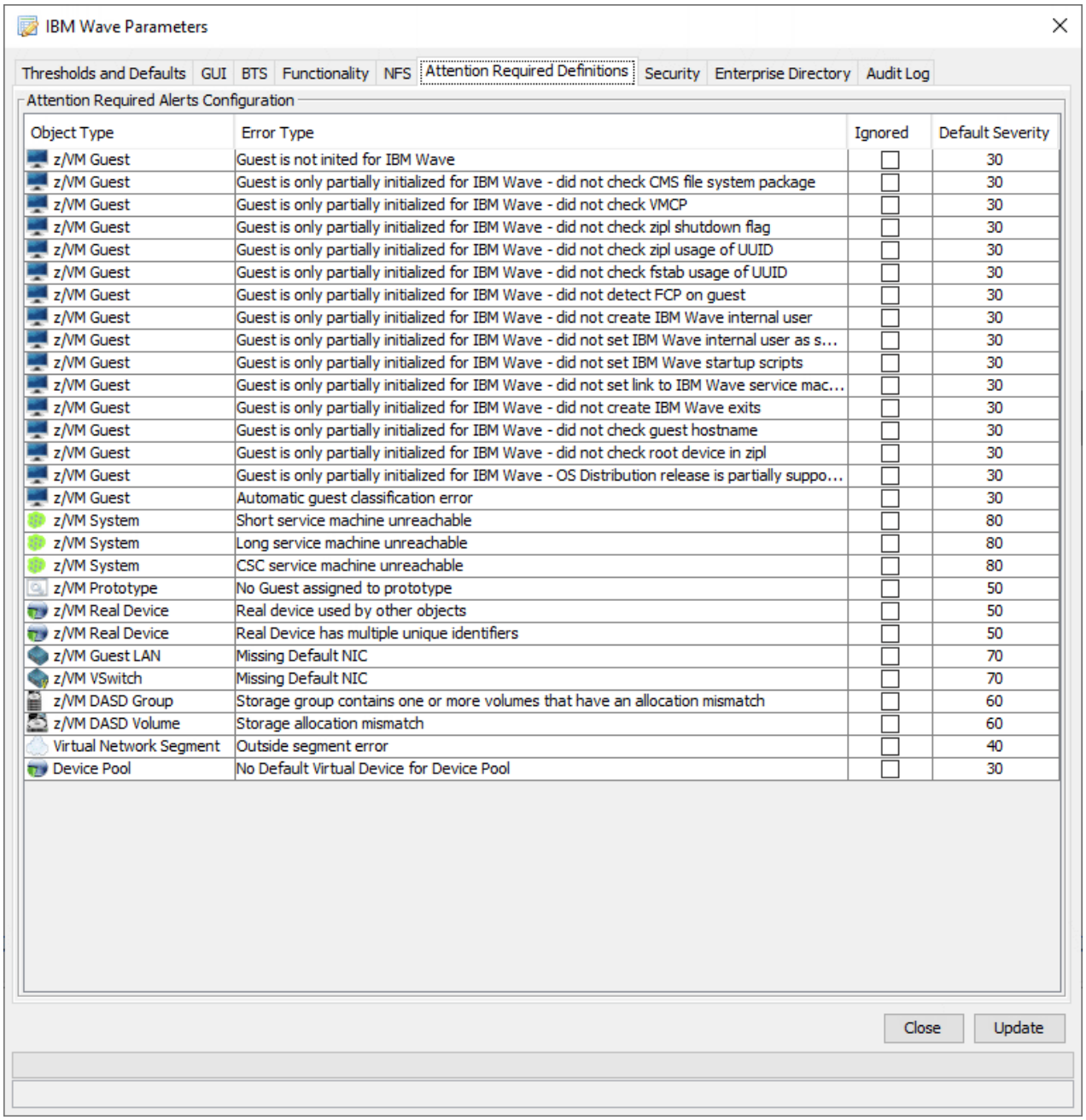


Figure 62. Attention Required Definitions

Security parameters

Access the security parameters from the IBM Wave main menu.

Click **Administrative** > **Manage Parameters** > **Security**.

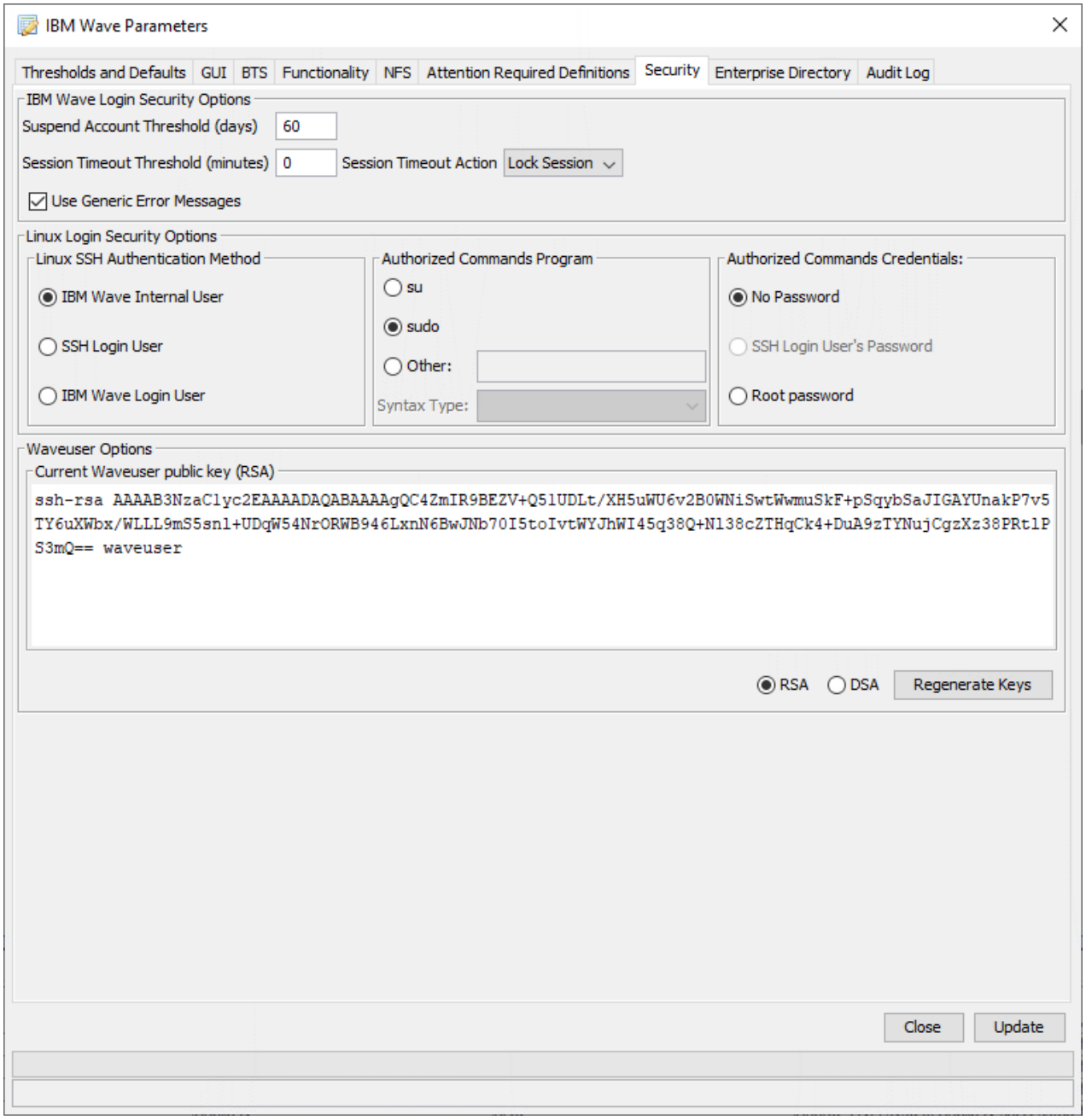


Figure 63. IBM Wave Parameters: Security

- **Suspend Account Threshold** - The number of inactive days required to suspend a user account automatically. The scheduled task to suspend users, **Update IBM Wave User Status**, is run every hour by default. For more information, see "BTS: Scheduling Tab" on page 107.
- **Session Timeout Threshold** - The number of inactive minutes required to time out the session.
- **Session Timeout Action** - The session timeout threshold. Currently, only **Lock Session** is supported. The session becomes locked and the IBM Wave user must reenter the login credentials to continue working with the application.
- **Use Generic error messages** - Use this checkbox to have IBM Wave issue generic error messages for a failed login attempt. A checked box is the default. Unchecking the box makes your installation less secure.
- **Linux SSH Authentication Method** - Use this setting to control Linux SSH authentication. For more information, see "Linux Login Security Options" on page 140.

- **Authorized Commands Program** - Use this setting to control how IBM Wave achieves root privileges when necessary. For more information, see “Linux Login Security Options” on page 140.

  Note: When you use a third-party security tool, such as ProdB, and enter the **su** command to achieve root authority, you must use the **Other** field and set the text to:

  ```
  full_path_to_pbrun/pbrun su
  ```

- **Authorized Commands Credentials** - Use this setting to control how the password is supplied when a user attempts to achieve root privileges. For more information, see “Linux Login Security Options” on page 140.
- **Waveuser Options** - Displays the public key that is defined for the authentication of the internal Linux user `waveuser`. This user is used to access Linux Virtual Servers during various IBM Wave functions and actions. You can use the **Regenerate Keys** action to regenerate a private/public key pair for `waveuser`.

  Note: The **Regenerate Keys** action marks all guests as not initialized for IBM Wave use and you should re-initialize the guests by using the **Init User for IBM Wave use** action. The private key is held encrypted in the IBM Wave database. For security purposes, IBM Wave does not provide a way to view the private key.

Enterprise Directory parameters

The Enterprise Directory parameters control the LDAP and Active Directory settings.

To access the Enterprise Directory parameters from the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Enterprise Directory**.

The options in the Enterprise Directory window can help optimize IBM Wave to your LDAP/Active Directory configuration.

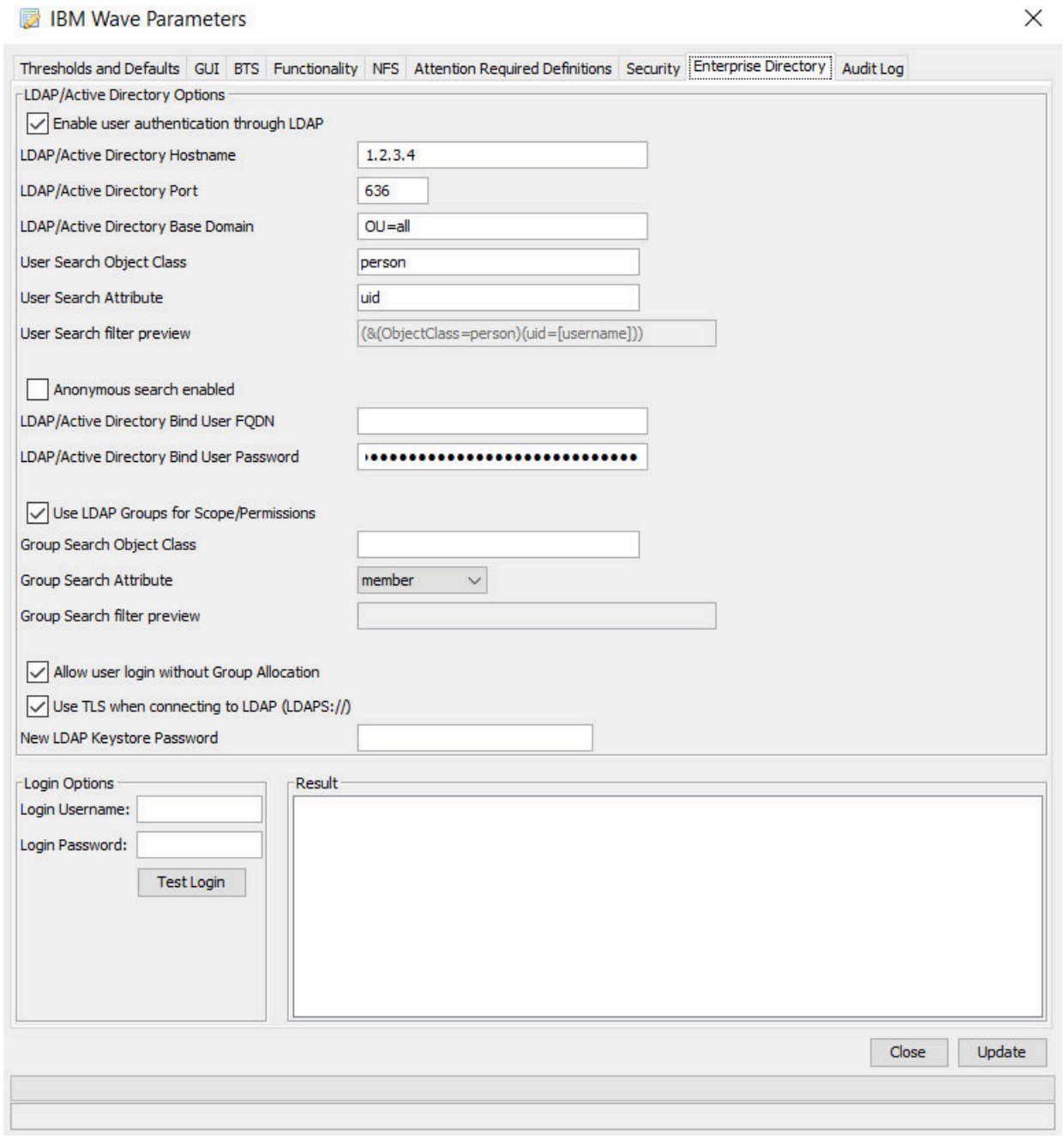


Figure 64. Enterprise Directory parameters

- **LDAP/Active Directory Hostname** - The host name or IP address of the LDAP authentication server.
- **LDAP/Active Directory Port** - The port on the LDAP authentication server on which to receive requests.
- **LDAP/Active Directory Base Domain** - The base domain name of the LDAP authentication server.
- **User Search Object Class** - The object class with which to search the LDAP directory for the specified user at login.
- **User Search Attribute** - The user attribute with which to search the LDAP directory for the specified user at login.
- **User Search filter preview** - Using the user object class and attribute provides a preview of the exact filter to be used to search for users in the LDAP directory. Verify that the setting is correct with your LDAP configuration.

- **Anonymous search enabled** - Indicates whether to attempt the initial bind and search against the LDAP server anonymously. When the field is disabled, to do the bind, the administrator must use the fully-qualified domain name (FQDN) and must provide the password.
- **LDAP/Active Directory Bind User FQDN** - The fully-qualified domain name of the bind user on the LDAP authentication server.
- **LDAP/Active Directory Bind User Password** - The password of the LDAP bind User.
- **Use LDAP Groups for Scope/Permissions** - When you select this field, user profiles in IBM Wave can be edited to include LDAP users based on their existing LDAP group membership. For more information about scope and permissions, see "Creating and updating IBM Wave user profiles" on page 155.
- **Group Search Object Class** - The group object class with which to search the LDAP directory for the groups the users belong to that are specified in the Group Scopes/Permissions pane.
- **Group Search Attribute** - The group attribute with which to search the LDAP directory for groups the users belong to that are specified in the Group Scopes/Permissions pane.
- **Group Search filter preview** - Using the group object class and attribute gives a preview of the exact filter that is used to search for user groups that belong to the LDAP directory. Verify that this setting is correct with your LDAP configuration.
- **Allow user login without Group Allocation** - Keep this check box selected if you want LDAP users without the proper scopes and permissions associated with the user group to log in to IBM Wave.
- **Use TLS when connecting to LDAP (LDAPS://)** - Selecting this option forces IBM Wave to authenticate to LDAP by using the SSL/TLS protocol. When enabled, you must specify the location of the SSL/TLS certificate or the directory that contains one or more certificates.
  - **New LDAP Keystore Password** - For LDAP authentication over SSL/TLS, you must store the certificates in a keystore on the WAVESRV server. Enter the keystore password in this field in order to change the password that is stored in the IBM Wave database. The new LDAP keystore password must match the current password of the keystore file. For more information, see Appendix K, "Using SSL and TLS certificates for LDAP or Active Directory login," on page 193

When you are ready to test your LDAP and Active Directory parameters, enter a **Login Username** and **Login Password** and then click **Test Login**.

Note: The **Login Username** and **Login Password** are used to test the parameters. They will not be saved between invocations of the **Manage Parameters** function.

Depending on the configuration you choose, IBM Wave sends the appropriate queries to test the configuration.

Audit Log parameters

The audit log parameters contain options for tailoring events that you want to audit and how IBM Wave presents and logs the events.

Access the Audit Log parameters from the **IBM Wave main menu**. Go to **Administrative** > **Manage Parameters** and click on the **Audit Log** tab. For instructions about using the Audit Log feature, see "Displaying audit log events" on page 159.

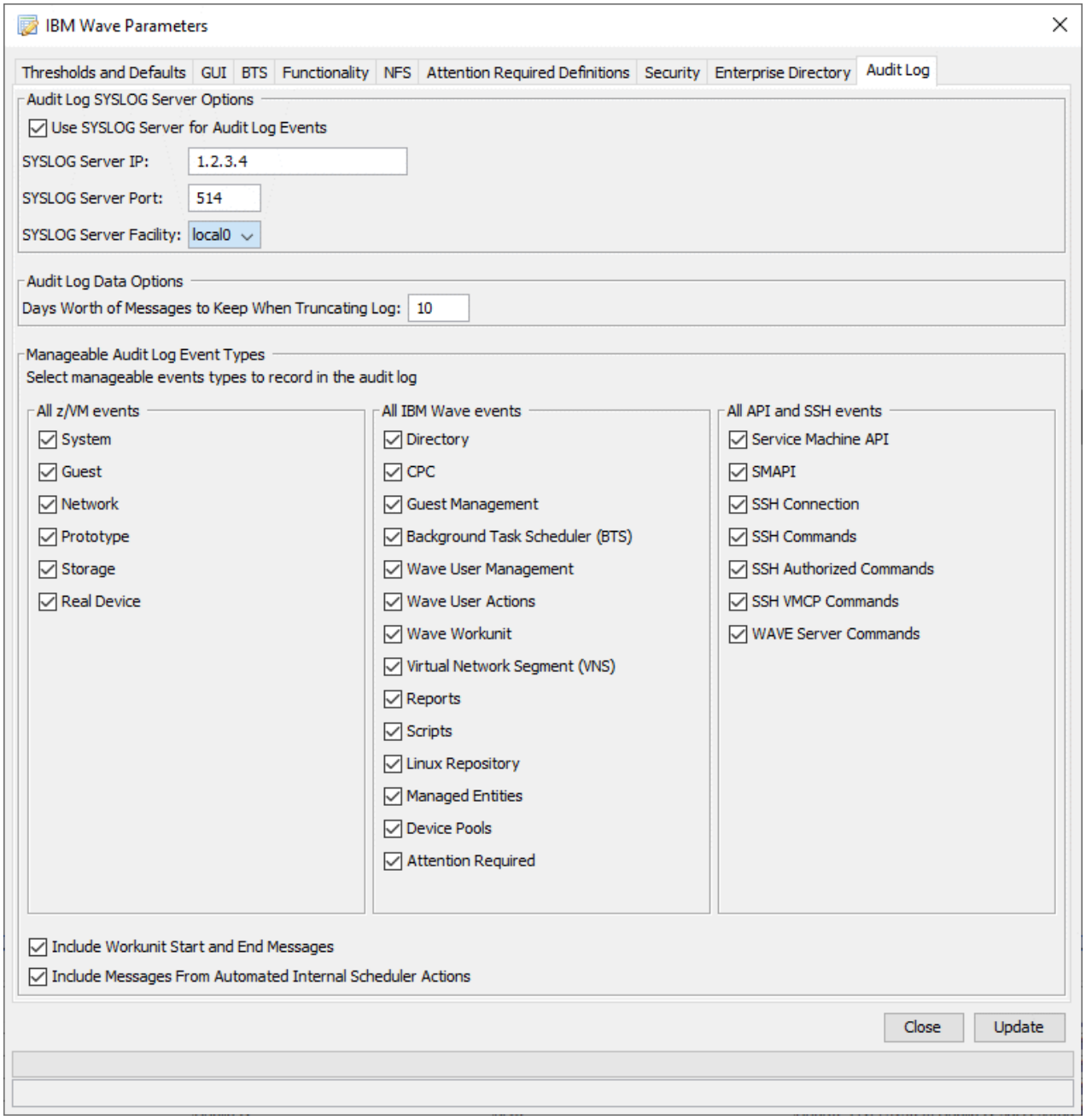


Figure 65. IBM Wave Parameters: Audit Log

The **Audit Log** tab contains the following options:

Audit Log SYSLOG Server Option

Use SYSLOG Server For Audit Log Events
Select to additionally use SYSLOG to record auditable events. When selected, IBM Wave routes all auditable events from the audit log to the SYSLOG host by using the server address, port, and facility information that you enter.

SYSLOG Server Host
The IP address of the server that hosts SYSLOG.

SYSLOG Server Port
The port number that is listening to collect the audit log events.

SYSLOG Server Facility
The local server name.

Audit Log Data Options

Days Worth of Messages to Keep When Truncating Log
The number of days worth of messages to keep in the audit log after you click **IBM Wave Audit Log Display > Actions > Truncate Log**.

Note: The default truncate value is one day (24 hours). To keep the file system size manageable, IBM Wave automatically truncates the log data every 24 hours and stores the file, `IBM-Wave-Log`*`YYYYDDD-HH:MM:SS`*, in the `tmp` file directory.

Manageable Audit Log Event Types

Note: By default, there are some events that can never be turned off. For example, security and IBM Wave parameters.

All z/VM Events
You can select or clear the following z/VM auditable event types:

- System
- Guest
- Network
- Prototype
- Storage
- Real Device

All Wave events
You can select or clear the following Wave auditable event types:

- Directory
- CPC
- Guest Management
- Background Task Scheduler (BTS)
- Wave User Management
- Wave User Actions
- Wave Work Unit
- Virtual Network Segment (VNS)
- Reports
- Scripts
- Linux Repository
- Managed Entities
- Device Pools
- Attention Required

All SSH and API Events
You can select or clear the following API and SSH auditable event types:

- Service Machine API
- SMAPI
- SSH Connections
- SSH Command
- SSH Authorized Commands
- SSH VMCP Commands

Include Workunit Start and End Messages
When selected all work unit start and end messages are recorded as auditable events.

Include Messages from Automated Internal Scheduler Action
This option is off by default. When selected, all internal messages are recorded as auditable events.

- For all of the topics that cover the audit log feature, see Chapter 8, “Audit Log Reporting feature,” on page 159.
- To control the **Audit Log Preview** options, see “GUI parameters” on page 116.

Changing User Preferences

You can change the user preferences to customize some of the features, including SSH, in IBM Wave for z/VM.

To change the **User Preferences**, from the **IBM Wave Main Menu**, click **User Tasks** > **Change IBM Wave User Preferences**.

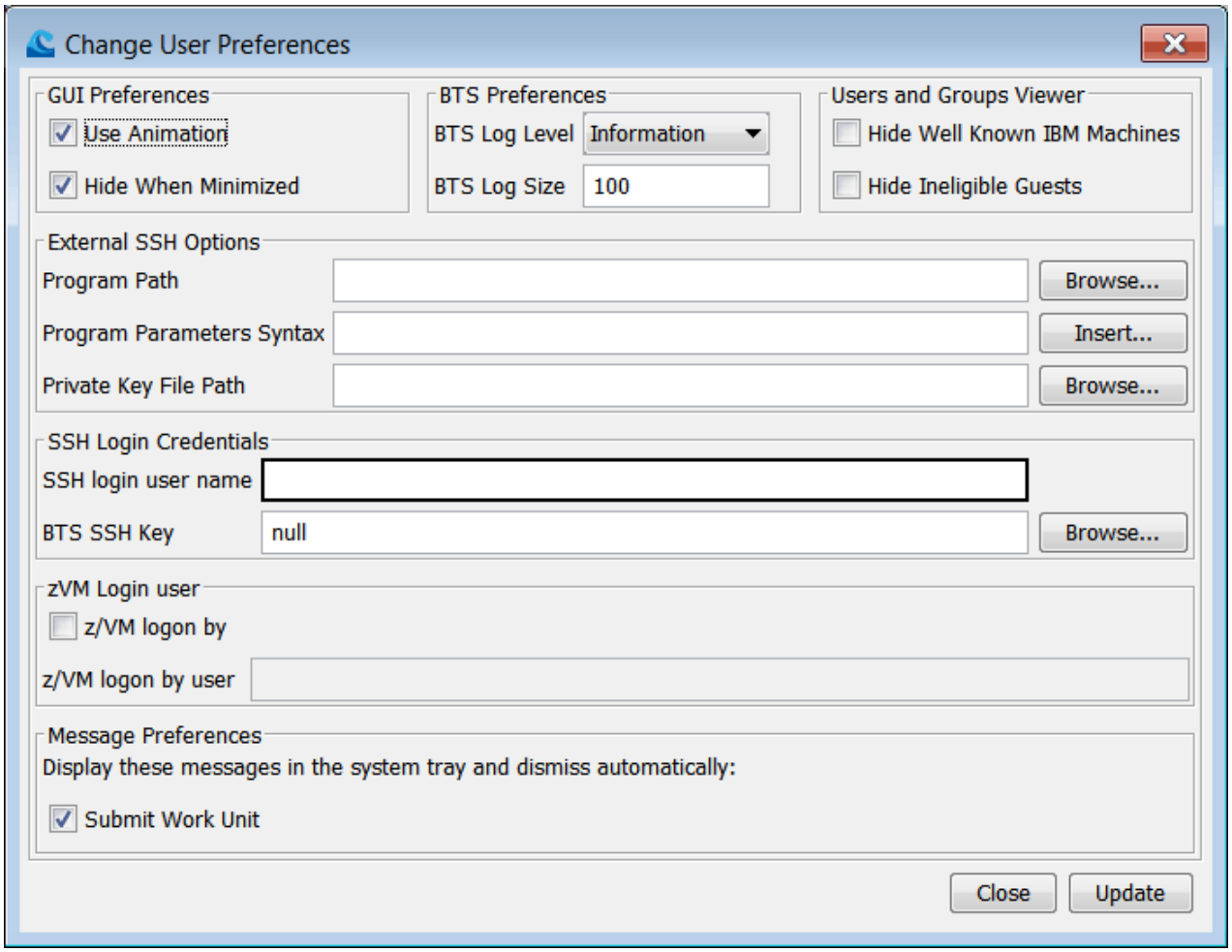


Figure 66. Change User Preferences

You can customize the following fields:

GUI Preferences

- **Use Animation** - This check box indicates whether the IBM Wave user interface uses animation whenever a new viewer is accessed, or when items are arranged in the viewer. Animation also applies to lay out, scroll, and zoom processing.
- **Hide When Minimized** - Use this option to control whether the IBM Wave application is minimized normally by showing a tab in the Windows taskbar. When you check **Hide When Minimized**, the tab is hidden. To access Wave, right-click on the **IBM Wave icon** in the Windows taskbar, and then select **Restore**.

BTS Preferences

- **BTS Log Level** - Indicates the level of messages to display in the Background Task Scheduler (BTS) Log viewer in the **General Status Viewer**.
- **BTS Log Size** - Indicates how many lines are saved in the BTS Log Viewer.

Users and Groups Viewer

- **Hide Well Known IBM machines** - Controls whether known machines are displayed in the **z/VM User Groups** and the **Network** Viewers in the **Current System Viewer**.
- **Hide Ineligible Guests** - Controls whether guests that are marked as ineligible for logon are displayed.

External SSH options

- **Program Path** - Use this option to define an external application to be started when a z/VM Guest is accessed by using SSH. Click "Browse" to locate the program. For more information, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_ssh_access.html.
- **Program Parameters Syntax** - Click **Insert** to automatically input the placeholder parameters.

  Select from the following placeholders for the external SSH application:

  – **Insert Login User** - Automatically inserts `LOGIN_USER` to provide a placeholder variable for the Linux user.
  – **Insert IP Address** - Automatically inserts `IP_ADDRESS` to provide a placeholder variable for the IP address.
  – **Insert Login Password** - Automatically inserts `LOGIN_PASSWORD` to provide a placeholder variable for the Linux user password.

    Important: IBM Wave issues a warning message every time this parameter is used.
  – **Insert Key File Location** - Automatically inserts `KEY_FILE_LOCATION` to provide a placeholder variable for the private key file location. For more information about the location, see **Private Key File Path**.
  – **Insert Key File Passphrase** - Automatically inserts `KEY_FILE_PASSPHRASE` to provide the placeholder variable for the key file passphrase.

    Note: Not all SSH programs can accept a passphrase in the command line.
  – **Insert default PuTTY Key Syntax** - Automatically inserts `Default PUTTY syntax` to provide the placeholder variables for the parameters that are required to run PuTTY with a user and key. For example, `<LOGIN_USER>@<IP Address> -i <KEY_FILE_LOCATION>`.
  – **Insert default PuTTY Password Syntax** - Automatically inserts `Default PUTTY syntax` to provide the placeholder variables for the parameters that are required to run PuTTY with a user and password. For example, `<LOGIN_USER>@<IP Address> -pw <LOGIN_PASSWORD>`.

    Note: The `LOGIN_PASSWORD` and `KEY_FILE_LOCATION` variables are mutually exclusive. They specify the method of user authentication that the external SSH application uses.
- **Private Key File Path** - The **Private Key File Path** preference can be used if the external SSH application is using Private/Public key authentication. The format of the private key file that you specify here must match the format that is expected by the external SSH application.

SSH Login Credentials

- **SSH Login User Name** - The **SSH Login User Name** preference is either optional or mandatory based on the Authentication Method settings in the **IBM Wave Parameters**. To use this preference, you must either:

  – Set a default SSH login user that is used by IBM Wave for direct SSH access to guests.
  – Run actions on guests that require Linux interaction.

  For more information about when this preference is optional or mandatory, see “Linux Login Security Options” on page 140.

- **BTS SSH Key** - Use this preference to set a default location for the SSH Key file that is used by IBM Wave for direct SSH access to guests, or when you must run actions on guests that require Linux OS interaction. The **BTS SSH Key** field is disabled when the IBM Wave Internal User is specified on the **IBM Wave Parameters** > **Security** tab.

z/VM Login User

- **z/VM Logon By** - This preference can be checked when the LOGON-BY authentication method is to be used to access z/VM Guests by using a 3270 emulator.

  Note: This setting is for convenience purposes. When you access a z/VM Guest by using a 3270, you can override this parameter.

- **z/VM Logon By user** - This preference contains the user with which IBM Wave attempts to login to the 3270 guest with the LOGON-BY authentication method.

  Note: This setting is for convenience purposes. When you access a z/VM Guest by using a 3270, you can override this parameter.

Message Preferences

- **Submit Work Unit** - This preference can be checked to automatically dismiss the "Submit Work Unit" messages. You can also dismiss the message on the "Work Unit submitted to BTS" message (as shown in Figure 67 on page 134). Check "Don't show this message again" and click "OK".

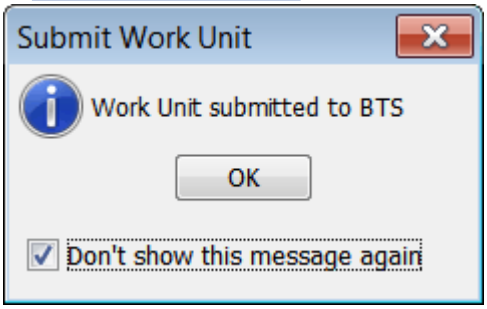


Figure 67. Dismiss Submit Work Unit messages

IBM Wave server options

This topic describes the customization options for the IBM Wave server.

- To understand how to modify the number of log files that are stored on the Wave server, see “Wave server log options” on page 134.
- When a new IBM Z system machine type or CPU ID does not match your existing z/VM system, see “Other Wave server options” on page 136.

Wave server log options

For log rotation to work, the standard Linux `cron` and `logrotate` programs must be installed on the Background Task Scheduler (BTS) Linux server. If either program is not installed, the Wave logs cannot automatically rotate, which means they can grow indefinitely.

You can modify the number of log files (`/var/log/WAVE/waveDebugLog.log*`) that are saved on the IBM Wave server by modifying the `logrotate` configuration file. This file, `logrotate.conf`, is stored in the `/usr/wave/install/` directory.

When the log file size is greater than 20 MB, log rotation occurs on an hourly basis, and a minimum of one week of log files are kept. The default for log rotation is "`rotate  168`" (24-hours X 7 days = 168).

To retain fewer log files, modify the `logrotate.conf` file. For example, if you change the value to "`rotate  48`", your installation keeps a minimum of two days of log files (24-hours X 2 days = 48).

Important: Currently, the recommended practice is to never change other parameters in the `logrotate.conf` file.

To change the daemon debug level for the BTS server, see “BTS parameters” on page 117.

IBM Wave API log options

The IBM Wave API uses a number of log files, as follows:

`/var/log/WAVE/APILog-`*`timestamp`*`.log` - These files contains all logging information for API calls. The amount of information that is collected in these logs can be set in `/usr/wave/API/user_config.properties` file.

The following parameters are supported.

log4j.appnder.file.MaxFileSize

Controls the size of the file. Default is set to 20 MB. This setting means that when the API log file reaches 20 MB the java logger creates a new file and stops writing to the current one.

log4j.appender.file.MaxBackupIndex

Controls the number of backup files that the logger keeps. When this number is reached, the oldest backup is deleted to make room for the new one.

log4j.rootLogger=INFO, file

Controls the logging level of the `APILog-`*`timestamp`*`.log` file. The word "file" is required to cause the log to be created. Options for the first parameter are derived from "log4j" logging options:

TRACE
The TRACE Level designates finer-grained informational events than the DEBUG.

DEBUG
The DEBUG Level designates fine-grained informational events that are most useful to debug an application.

INFO
The INFO level designates informational messages that highlight the progress of the application at coarse-grained level.

WARN
The fileWARN level designates potentially harmful situations.

ERROR
The ERROR level designates error events that might still allow the application to continue running.

FATAL
The FATAL level designates severe error events that presumably leads the application to abort.

WebSphere Liberty log files

The WebSphere Liberty log files reside in `/usr/wave/websphere/wlp/usr/servers/defaultServer/logs`. The directory contains the following files:

console.log
This file contains the redirected standard output and standard error streams from the Java virtual machine (JVM) process.

messages.log
This file contains all messages that are written or captured by the logging component. All messages that are written to this file contain additional information such as the message time stamp and the ID of the thread that wrote the message. This file does not contain messages that are written directly by the JVM process.

Controlling the WebSphere Liberty logs is done through the `server.xml` file using the following properties:

consoleLogLevel

This filter controls the granularity of messages that go to the `console.log` file. The valid values are INFO, AUDIT, WARNING, ERROR, and OFF. By default, the level is OFF.

maxFileSize

The maximum size (in MB) that a log file can reach before it is rolled. The WebSphere Liberty runtime only rolls logs based on size. To disable this attribute, set the value to 0. The maximum file size is approximate (that is, the log might be slightly larger to enable a complete API call to be recorded). By default, the value is 20.

Note: This property does not apply to the `console.log` file.

maxFiles

If an enforced maximum file size exists, this setting is used to determine the number of log files that are kept. This setting also applies to the number of exception logs that summarize exceptions that occurred on any particular day. So if this number is 10, you might have 10 message logs, 10 trace logs, and 10 exception summaries in the FFDC directory. By default, the value is 2.

Note: This property does not apply to the `console.log` file.

For more information, see Logging and Trace.

Other log files and directories

trace.log

This file contains information that is generally logged in `messages.log`, but the `server.xml` file indicates that this information should be hidden.

FFDC

This directory contains exceptions that are logged individually.

Other Wave server options

Use the following procedure when the machine type (8562, for example) or CPU ID for a new IBM Z system is not one listed in “IBM mainframe requirements” on page 1. This procedure will prevent the auto-detect work unit from failing when IBM Wave does not recognize a system.

Create a blank file named "disableCPCchecking" in the following directory in Wave server file system: `/usr/wave/config`.

Chapter 6. Security

IBM Wave contains an integrated security subsystem. All of the security data is kept encrypted in the IBM Wave database on the WAVESRV server.

IBM Wave supports Secure FTP. Secure FTP is used by default, unless IBM Wave detects that your installation is using standard FTP.

To understand all aspects of IBM Wave security, review the following topics:

- "Security parameters" on page 125.
- "IBM Wave security tasks" on page 137.
- "Linux Login Security Options" on page 140.
- "The password resetter utility" on page 142
- "Disabling Wave server certificate validation in the IBM Wave client" on page 142.
- "IBM Wave user authentication" on page 143 includes the following topics:
  - "IBM Wave user profiles" on page 144
  - "LDAP group-based security" on page 144
  - "3270 SSL/TLS support" on page 144
- Chapter 7, "User management," on page 147

IBM Wave security tasks

This section contains some information about, and pointers to, IBM's recommendations for how various types of users can make sure they are deploying IBM Wave in a secure manner.

Wave server Linux administrator tasks

Security tasks for Wave server Linux administrators include the following:

- Configure ports.

  To change the port number *during installation*, use the procedure in step "8" on page 64. To change the port number *when updating Wave* (to a new fix pack, for example), use the procedure in "Upgrading IBM Wave for z/VM" on page 68.

  See "Port reference information" on page 77.
- Change WebSphere Liberty's keystore password, which is used to secure the private key for the Wave server, during installation, and periodically.

  See "Changing a keystore password" on page 200.
- Replace WebSphere Liberty's default self-signed certificate with a valid one signed by a CA that client workstations will trust.

  See Appendix L, "Managing Wave's server certificate," on page 195.
- Add CA signing certificates to trust stores for cases in which Wave needs to validate another server's certificate, as follows:
  - Java on the Wave server for the z/VM systems that are being managed.

    See Appendix J, "Configuring certificates for managed z/VM systems," on page 191.
  - Java on the Windows workstation for 3270 and CLC connections for the z/VM systems.

    See Appendix J, "Configuring certificates for managed z/VM systems," on page 191.
  - LDAP or Active Directory for Wave login.

See Appendix K, “Using SSL and TLS certificates for LDAP or Active Directory login,” on page 193.

Network administrator tasks

Security tasks for network administrators include the following:

- Configure firewalls.

  See:

  – “Port reference information” on page 77
  – “Firewall information” on page 80.

Wave application administrator tasks

Security tasks for Wave application administrators include the following:

- Change the Wave database password during installation, and periodically.

  See “Regenerate IBM Wave Database Password” on page 101.

- Update the Wave GUI's security parameters.

  From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **GUI**:

  – Check the box under **GUI to BTS communication Security** to **Use TLS encryption for communication between BTS and GUI**.

  See “GUI parameters” on page 116.

- Define all Wave users (including at least one Wave administrator) in an enterprise directory so their passwords are managed according to your enterprise's password policy.

  See “IBM Wave user authentication” on page 143.

- Update configuration options.

  From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Functionality**:

  – Uncheck the box to accept all certificates.

  See “Functionality parameters” on page 119.

- Set security parameters.

  From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Security**:

  – Make sure an account suspension threshold is set.
  – Make sure a non-zero session timeout is set.
  – Check the **Use Generic error messages** box.

  See “Security parameters” on page 125.

- Review security parameters.

  From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Security**:

  – Review the Linux login security options.
  – Regenerate keys periodically.

  See “Security parameters” on page 125.

- Set enterprise directory parameters.

  From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Enterprise Directory**:

  – Check the box to encrypt connections to the enterprise directory server.
  – Set the LDAP keystore password.

  See “Enterprise Directory parameters” on page 127.

- Set audit log parameters.

From the IBM Wave main menu, click **Administrative** > **Manage Parameters** > **Audit Log**:

- Check the box to send audit log events to an external SYSLOG server.

See "Audit Log parameters" on page 129.

- Review audit log parameters.

From the IBM Wave main menu, click **User Tasks** > **View Audit Log**:

- Periodically review all warning and error audit records.

See "Displaying audit log events" on page 159.

- For each managed z/VM system, do the following:
  - Update the system properties.
  - Check the `Encrypt with TLS` box.
  - Check the `Validate server identity` box.
  - Check the `Use TLS tunnel` box.

See Update details.

- Start the Wave GUI.

Expected result: You will see a green `Secure connection` message.

See "Start IBM Wave for z/VM" on page 66.

Wave client workstation administrator tasks

Security tasks for Wave client workstation administrators include the following:

- Make sure the CA who signs the Wave server's certificate configured by the Wave server Linux administrator on the Wave server is in the trust store of the Java workstation, unless the Wave server Linux administrator has manually disabled certificate validity checking in the Wave GUI.

See Appendix J, "Configuring certificates for managed z/VM systems," on page 191.

- Review the following topics if you are using 3270 or CLC access.

See:

- "3270 SSL/TLS support" on page 144
- Appendix J, "Configuring certificates for managed z/VM systems," on page 191.

z/VM administrator tasks

Security tasks for z/VM administrators include the following:

- Configure z/VM's TCP/IP stack to use Transport Layer Security (TLS).

See Setting up for SSL/TLS.

- Configure z/VM's TCP/IP stack with a valid certificate signed by a certificate authority (CA) that the Wave server and client workstations will trust.

See Enabling SSL/TLS Support.

- Configure the FTP server.

See Configuring the FTP Server.

- Configure ports.

See "Port reference information" on page 77.

- Configure the Wave service machines.

See "Configuring IBM Wave service machines " on page 57.

- Create the Wave service machines.

See “Creating the service machines” on page 57.

– Create grants for persistent LANs.

  See “Review the parameter files” on page 30.

Auditing

You can use IBM Wave's Audit Log Reporting feature to audit actions from z/VM systems and managed guests, z/VM API (IBM Wave service machine and SMAPI) events, and Secure Shell (SSH) events.

See Chapter 8, “Audit Log Reporting feature,” on page 159.

Diagnosis

This section includes information to help you diagnose possible problems that could occur while administering IBM Wave.

See Diagnosing your connection status.

Linux Login Security Options

Administrators can determine how Linux authorization gets implemented for the IBM Wave installation. To change the default Linux login security options, from the **IBM Wave main menu**, select **Administrative** > **Manage Parameters** > **Security** and go to the "Linux Login Security Options" pane.

In general, IBM Wave takes Linux actions for guest-specific operations. Some of the following actions drive Linux flows:

- **Init Users for IBM Wave**

  Note: The IBM Wave Internal User, WAVEuser, is installed on each Linux guest that is managed by Wave during the **Init Users for IBM Wave** action. The Wave Internal User can be used by the WAVESRV server to connect to the Linux Guest only by using the public/private key pair. The password for WAVEuser is locked, which means the user cannot log in by using a password.
- **Manage Storage Actions**
- **Connect to VNS** or **Disconnect from VNS**.

The Linux flows establish a Secure Shell (SSH) connection to the managed guest, and then run the Linux commands on the guest. When the flows are run, some of the necessary Linux commands require the use of the **su** or **sudo** command for authorization. For example, during an **Init Users for IBM Wave** action, you must use **sudo** to add a user to your managed guest.

As an administrator, you can use the "Linux Login Security Options" pane to control the flows and determine how Linux authentication and authorization are implemented for your installation.

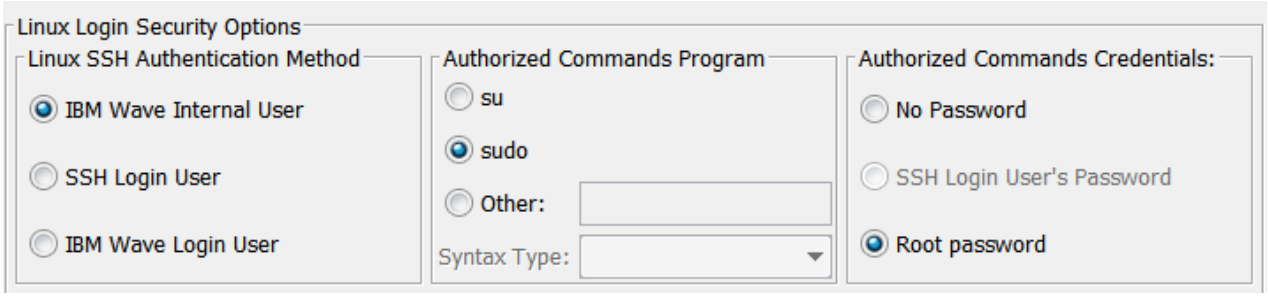


Figure 68. Linux Login Security Options

Linux SSH Authentication Method

The **Linux SSH Authentication Method** options determine what user security options are used when IBM Wave runs commands on the Linux guest. The user can also be configured with the **sudo** command. When the configuration uses **sudo**, the user must be listed in the `/etc/sudoers` file (which you access by using the **`visudo`** command).

- "IBM Wave Internal User" (WAVEuser): Select "IBM Wave Internal User" when everyone on the designated Wave server is running commands on all of the managed guests by using the WAVEuser Linux ID that is installed on each Linux guest during the **Init Users for IBM Wave** process.
- "SSH Login User": Select the "SSH Login User" when everyone on the designated Wave Server is running commands on all managed guests by using a user-designated (not site-wide) login ID. The ID must have a default value assigned. To assign the default value, on the **IBM Wave main menu**, click **User Tasks > Change User Preferences > SSH login user name**. The SSH user must be manually created on each managed Linux guest and configured with a home directory. When **sudo** is selected, you must add the user to the `/etc/sudoers` file.
- "IBM Wave Login User": Select "IBM Wave Login User" when everyone on the designated Wave server is running commands on all the managed guests by using their IBM Wave user interface (UI) ID. The user must be manually created on each managed Linux guest such that the new Linux user name matches the IBM Wave UI login name. The user must be configured with a home directory. When **sudo** is selected, you must add the users to the`/etc/sudoers` file. This configuration is common when the Lightweight Directory Access Protocol (LDAP) is used on IBM Wave and the Linux guests.

Authorized Commands Program

The **Authorized Commands Program** determines the authorized commands that are run on Linux.

- **su**: The **su** option uses **su** to run commands on Linux. You must select "Root password" in the **Authenticated Access Using** pane for your **Linux Authentication Method** user to automatically switch to root when they run commands that need authorization.
- **sudo**: The **sudo** commands run authorized on Linux, but it does not switch the user. Commands can be run by the specified **Linux Authentication Method** by using the **sudo** command.
- **`Other`**: The **`Other`** option is for customers who use Enterprise Security Managers on Linux to allow them to enter an alternative **su** or **sudo** commands.
  - **`Syntax Type`**: The **`Syntax Type`** field indicates whether the command syntax specified matches **su** or **sudo**.

Authorized Command Credentials

Authorized Command Credentials determines how the **Authorized Commands** you select (su, sudo, Other) are configured.

No Password
Use when no password is needed when issuing an authorized command.

SSH Login User's Password
Use when the SSH user's password is needed when issuing an authorized command.

Root password
Use when your `sudoers` file requires the root password when issuing an authorized command.

Typically you use "Root password" with **su** and "SSH Login User's Password" with **sudo**".

Sometimes, **sudo** is configured not to prompt for passwords at all. With this **sudo** configuration, you can use the "No Password" option to instruct IBM Wave not to prompt the user for an authorized command password.

Other times, **sudo** is configured to use the "Root password" to authorize the commands. When you SSH into your system as a regular non-root user and issue a command (for example, **sudo `cat /etc/passwd`**), Linux can either request your user password or the root password. Make your selection according to the command output.

Notes:

1. To manually configure **sudo** to use the root password, enter the "Defaults rootpw" line into the `sudoers` file (**`visudo`**).
2. To manually configure **sudo** to use the user password, remove the line from the `sudoers` file that either says "Defaults rootpw", "Defaults runaspw", or "Defaults targetpw".

For more information about the IBM Wave Security parameters, see “IBM Wave parameters” on page 113.

The password resetter utility

The topic describes when you might need the password resetter utility, how to access it, and its parameters.

Before you begin

The password resetter utility performs one of the following functions, depending on its input parameters:

- Resets the password, the security question, and the security answer for a single user defined in the IBM Wave database.
- Expires the passwords of all users defined in the IBM Wave database whose passwords do not comply with IBM Wave's password policy.

IBM recommends that you create more than one IBM Wave administrator-type user who is allowed to manage other administrator-type users. Circumstances can arise when a user is revoked because of the following reasons:

- Login attempts are too frequent.
- Lapses in security policies prevent the passwords from being kept current.

To help remedy these situations, IBM Wave includes a password resetter utility called **WAVEPasswordResetter** that the Wave server's Linux administrator can use.

About this task

The **WAVEPasswordResetter** command is in the `/usr/wave/WAVEBackground` directory.

Procedure

- Run any of the following variations of the **WAVEPasswordResetter** command:

  WAVEPasswordResetter [-e]
  Expires all users whose passwords do not comply with IBM Wave's password policy, which prohibits the use of semicolons and requires that passwords be 8-32 characters in length.

  WAVEPasswordResetter [-h]
  Displays usage information.

  WAVEPasswordResetter [-u *user_name*]
  Resets a single user's password. When you specify the **-u** option to change the password, as indicated in the resulting prompt, if *user_name* is logged in when the administrator changes the user's password, *user_name* is logged off immediately.

Related reference

“WAVEPasswordResetter command” on page 202

Disabling Wave server certificate validation in the IBM Wave client

If the Wave server Linux administrator is unable to obtain a valid certificate identifying the Wave server signed by a certificate authority that workstations running the IBM Wave client will trust, this administrator can disable certificate validation in the application. When using an encrypted connection, IBM Wave requires a valid certificate, unless the administrator disables validity checking manually.

To disable certificate validation in the application, the Wave server Linux administrator edits the `/usr/wave/GUI/WAVE.jnlp` file. This example shows the final lines of this file, as supplied by IBM:

```
    <!-- 3300 is the default port.  If your Wave admin chooses a different port AND
you uncomment this code, update the port too.
    <argument>3300</argument>
    <argument>60000</argument>
```

```
    <argument>trustanybtscert</argument>
     -->
  </application-desc>
</jnlp>
```

This example shows the final lines of the `/usr/wave/GUI/WAVE.jnlp` file, as edited to disable server certificate validity checking (the extra <argument> elements are "uncommented"):

```
    <!-- 3300 is the default port.  If your Wave admin chooses a different port AND
you uncomment this code, update the port too.
     -->
    <argument>3300</argument>
    <argument>60000</argument>
    <argument>trustanybtscert</argument>
  </application-desc>
</jnlp>
```

After the administrator has disabled certificate validation (or later reverses the process to enable it), IBM Wave client users must clear their Java application caches and browser caches in order to observe the change. See Diagnosing your connection status for the resulting IBM Wave client changes.

Users of client workstations running IBM Java can clear their Java application caches by running **Configure Java**, navigating to the **General** tab of that application, clicking on the **Settings...** button under **Temporary Internet Files**, clicking on the **Delete Files...** button, and selecting *both* Cached and Installed applications and applets. Users running other supported Java versions should consult their JVM's documentation for the equivalent procedure. Users should follow their browser's documented procedure for clearing any cached data related to IBM Wave's server.

IBM Wave user authentication

IBM Wave user authentication, password checking, is done in one of the following ways:

- Database authentication - The IBM Wave user password is saved in the IBM Wave database. During login processing, the user's password is retrieved from the IBM Wave database, and compared to the password that the user entered. The passwords are encrypted with the Advanced Encryption Standard (AES) algorithm. For complete instructions, see "IBM Wave Database Actions" on page 99.
- Enterprise Directory authentication - The IBM Wave user password is saved in the LDAP/Active Directory database. During login processing, the user's password is checked against an entry in the site's LDAP/Active Directory database. LDAP/Active Directory authentication requires that certain IBM Wave parameters be set. The parameters can be viewed, updated, and tested in **Administrative** > **Manage Parameters** > **Enterprise Directory**.

Note: For database authentication, IBM Wave's password policy prohibits the use of semicolons and requires that passwords be 8-32 characters in length.

If Wave's password policy is sufficient for your installation's needs, you can define Wave users as local/database users, or you can define some users in the enterprise directory, or both.

Otherwise, define all Wave users (including at least one Wave administrator) in an enterprise directory (ED) so their passwords are managed according to your enterprise's password policy. After confirming that at least one ED-defined user can authenticate and deleting the initial user defined during installation, Wave allows you to suspend or delete all users defined in the IBM Wave database whose passwords might not comply with your enterprise's requirements.

The potential risks follow:

- If you delete the last/only database user, you're completely dependent on the enterprise directory connection to keep working, and changing its configuration requires an administrator (though not a super user). This could pose a denial of service risk. Reinstallation would be required, for example, if the enterprise directory server became permanently inaccessible using the connection information configured in Wave.

- If you suspend the last/only database user ID instead of deleting it, you must trust the Wave server's Linux administrators to not reset its password and thus acquire the role of Wave application administrator, except in such emergencies as the example described previously.

IBM Wave user profiles

Some of the security definitions for IBM Wave Users are defined in profiles. These profiles contain:

- Role definitions
- User type definitions
- Scope/permission sets.

By default, whenever an IBM Wave user entry is created (manually or automatically), a corresponding profile is created. These profiles are managed internally by IBM Wave, and cannot be directly modified through the IBM Wave User Profile Manager. Rather, they are altered when updating the definitions through the IBM Wave User Manager.

IBM Wave User Profiles can also be manually created, modified, and removed through the IBM Wave User Profile Manager. There, profiles can be associated with LDAP groups (by name) and can be used to automatically assign roles, user types, and scope/permission sets to IBM Wave users logging in using LDAP authentication.

Generally speaking, when an IBM Wave user logs into the system, the role, user type, and scope/permission sets are assigned based on the ones specified in the profile or profiles assigned to the user.

LDAP group-based security

When using LDAP Authentication, security definitions for your users can be assigned to an LDAP group.

The LDAP assignment is done by using the **Manage User Profiles** action with security definitions (Roles, User Types, and Scope/Permission Sets), and then associating the profile with LDAP Group names.

When a user logs in to IBM Wave, the LDAP server is queried for the list of LDAP Groups to which the user belongs. IBM Wave then attempts to match profiles to the names of the groups.

After all the profiles are located, they are compiled together to generate the user role, type, and scope and permission sets.

Conflicts are handled in the following manner:

- **Role Conflicts** - Role definitions in IBM Wave User Profiles are either "Granted" or "Not Set". Denying a role is not permitted. If, for example, a user is assigned to two profiles, one specifies the SLA role and the other the NWA role, the combined security context for the user is SLA and NWA.
- **User Type Conflicts** - The "Administrator" user type supersedes the "Regular" user type. If, for example, a user is assigned to two profiles, one specifying the "Administrator" user type and the other the "Regular" user type, the user is assigned the "Administrator" user type.
- **Scope and Permission Conflicts** - Permissions for actions in a particular scope can be defined as either "Granted", "Not Set", or "Denied". If two profiles are defined for the user with the same scope, permission conflicts are resolved in the following manner:
  - Not Set versus Granted - The permission is granted.
  - Not Set versus Denied - The permission is denied.
  - Not Set versus Not Set - The permission is denied.
  - Granted versus Denied - The permission is denied.

3270 SSL/TLS support

IBM Wave supports two methods of conducting SSL/TLS sessions for 3270:

1. **Start TLS** - The **Start TLS** method specifies that the actual 3270 handshake and communication uses SSL/TLS.

2. **SSL Tunneling** - The **SSL Tunneling** method specifies that the 3270 communication is done normally, but under an SSL/TLS tunnel.

The SSL tunneling option must be configured per z/VM System in the z/VM System parameters. For more information on the z/VM System Parameters, see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/ug_display_details.html.

Regardless of the SSL/TLS method selected, when using SSL the z/VM System's SSL certificates must be imported into the Java Keystore on the workstation, or the "Accept All Certificates" parameter must be set in the IBM Wave parameters.

- For information about how to import an SSL/TLS certificates, see Appendix J, “Configuring certificates for managed z/VM systems,” on page 191.
- For more information about the IBM Wave parameters, see “IBM Wave parameters” on page 113.

Chapter 7. User management

An administrator can add, update, and manage users through a set of administrative actions in the IBM Wave client. Using the IBM Wave client, an administrator can manage user types, roles, assign scopes and permissions, update user profiles, change preferences for IBM Wave users, and use the password resetter utility.

Linux system administrators who manage Linux virtual servers are defined as IBM Wave users, with specific scopes and permissions.

If your installation enforces separation of roles and duties, use the following role definitions:

- Wave server Linux administrator
- Network administrator
- Wave application administrator
- Wave client workstation administrator
- z/VM administrator

For full separation of roles and duties, assign different people to each role. In some cases, you might find it simpler to assign the role of Wave server Linux administrator and the role of Wave application administrator to the same person, if your enterprise's security policies permit doing so.

Understanding user types and roles

This topic describes how users and roles work in IBM Wave.

Use the **Manage IBM Wave User** option to partition the tasks and resources that are permitted to each IBM Wave user as required by your site. To access this option, from the IBM Wave main menu, click **Administrative** > **Manage IBM Wave User**.

User types

Each IBM Wave user has a defined role that provides levels of permission with authority to allow or block certain tasks from being performed. Valid user types follow:

- Superuser – An all-inclusive scope with full permission for all current and future defined z/VM systems. The Wave server Linux administrator creates the superuser as part of installing IBM Wave (see step “9” on page 64 for more information). The superuser is a derivative of an administrative user and its purpose is to create definitions for all other necessary IBM Wave administrators. After all of the IBM Wave administrators are defined to manage individual z/VM systems, the superuser is no longer needed. So, unless you want to retain a user with this level of authority, suspend the superuser after more users are created.
- Administrator or IBM Wave administrator - The administrator role must be assigned to personnel who manage IBM Wave (by defining new users and systems and performing maintenance and other tasks). IBM Wave administrator roles include the following:
  - Wave server Linux administrator
  - Wave application administrator
  - Wave client workstation administrator.

  The standard recommendation is to define at least two administrators to avoid any issues with password expiration.
- **Regular user** – The regular user role is assigned to Linux system administrators who use IBM Wave for provisioning and management.

Roles

Each role has a set of base functions that cannot be added or removed by permissions, but can vary according to the scope. Users can be assigned one or more roles as necessary. Valid types of roles follow:

- Network administrator (NWA) – The only user in the system who can create, delete or update the network configuration (such as GLANs, VSwitches, and other network details). Actions on objects or elements that are not in the scope, such as VNS, VLANs and other elements, cannot be limited based on scope or permissions.
- Site-level administrator (SLA) – This role is similar to **root** access, but this user cannot interact with the network unless that role is also defined. The SLA must be the best one with the correct view of the IT and how IBM Wave manages it. The SLA is the only user that sees and manages the storage and the system views. Like the NWA, actions against elements and objects that are not the scope, such as DASD volumes, devices, and others, cannot be limited based on scope or permissions.
- None – This role has no base function.

IBM Wave supports the following user role combinations:

- Administrator with NWA and SLA authority
- Administrator with NWA authority (no SLA)
- Administrator with SLA authority (no NWA)
- Regular user with NWA authority
- Regular user.

Overview of scopes and permissions

Understand how scopes and permissions work in IBM Wave.

Each IBM Wave User can be assigned multiple **Scopes**. **Scopes** define the objects with which a user can view and interact. Each Scope contains a permission entry that describes the actions that the user can take on the objects within that Scope. IBM Wave always uses the most discrete Scope available for an object to assess the permissions.

For example, if a user has the following scopes:

- One Scope for all z/VM Guests belonging to **System A, Project *** with all permissions. The asterisk ("*") assigns all projects.
- Another Scope is defined for **System A, Project A** with no permissions, which is more discrete than when all permissions given.

Even though the user has permission to all projects, because IBM Wave assigns the most discrete scope, the user cannot take any actions on z/VM Guests belonging to **System A, Project A**.

IBM Wave contains the following **Scope Types**.

1. **Systems** – The z/VM systems and LPARs that the user can manage.
2. **Projects** – The guest actions that the user can manage.
3. **DASD Groups** – The direct access storage device (DASD) and storage actions that the user can manage.
4. **Device Pools** – The device pool resources that the user can manage.

In summary, a user can have authority to a **scope** of a particular set of resources within their **role**.

Permissions

Permissions define the set of tasks that a user can run. An administrator can define tasks with the following permissions:

- Permitted
- Blocked

- Not available.

Tasks differ based on the type of managed object. IBM Wave contains the following types of **Permissions** to managed objects that you can assign to users

z/VM System Permissions

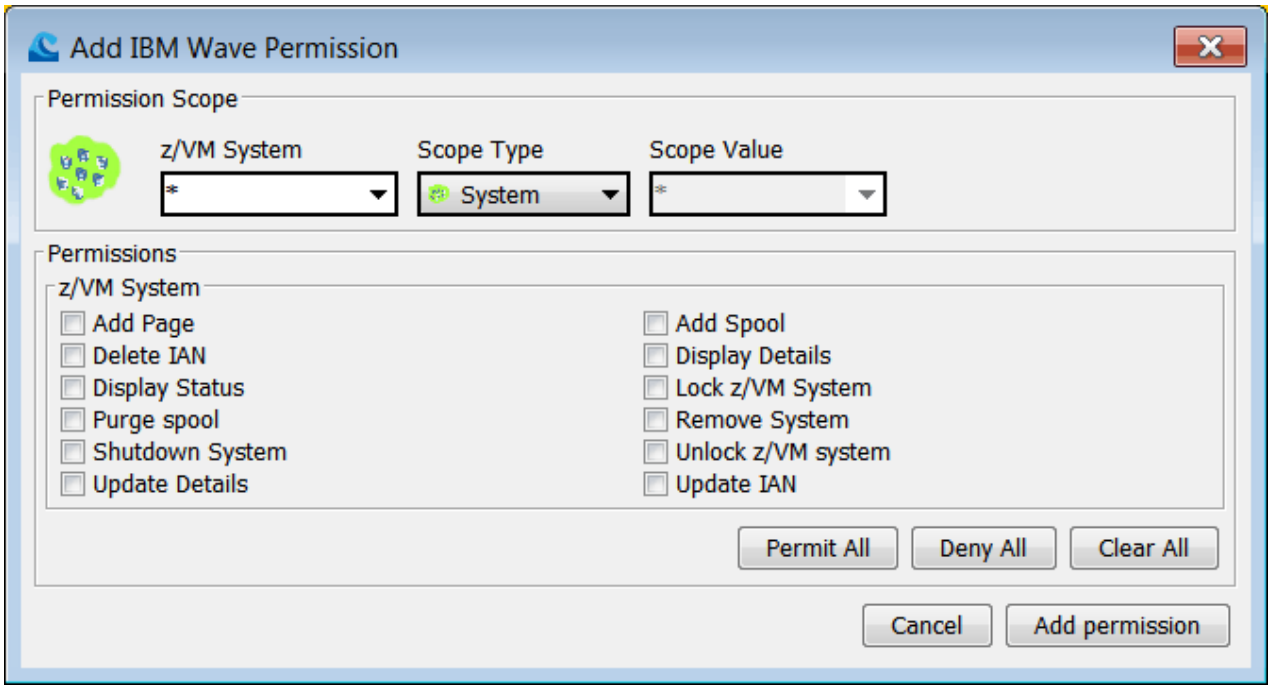


Figure 69. z/VM System Permissions

Project Permissions

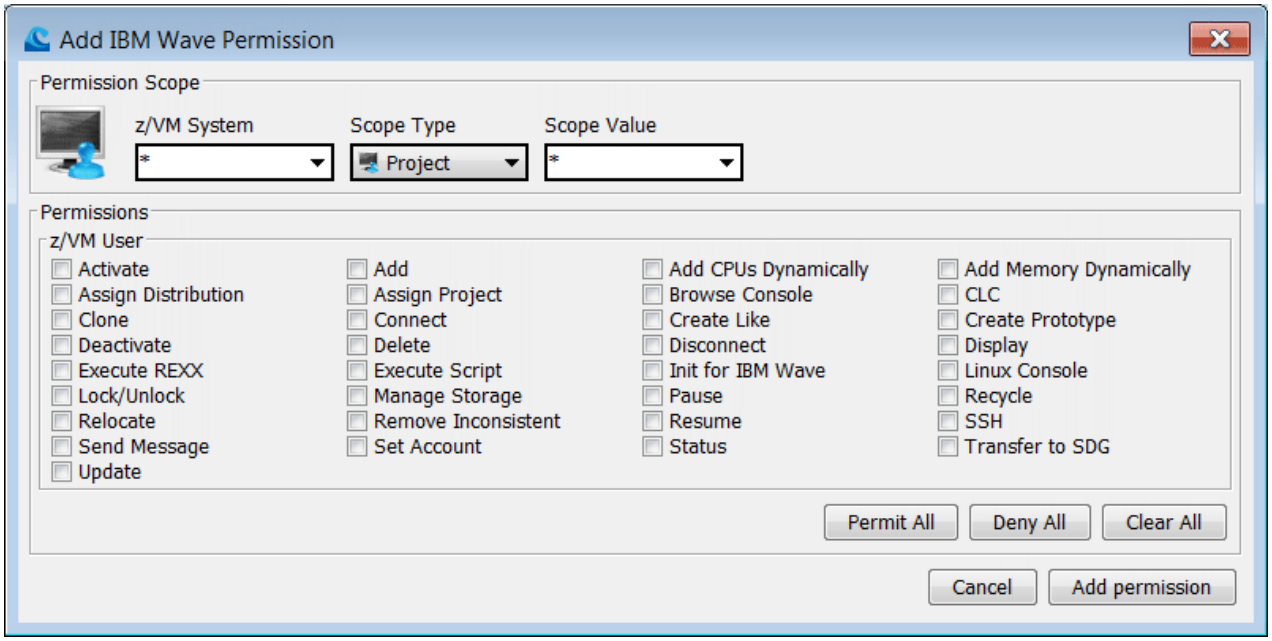


Figure 70. Project Permissions

Important: By default, every user who has the **activate** and **deactivate** permissions also has the **relocate** permission. This default is true even when the only permissions the user has are activate and deactivate.

DASD Group Permissions

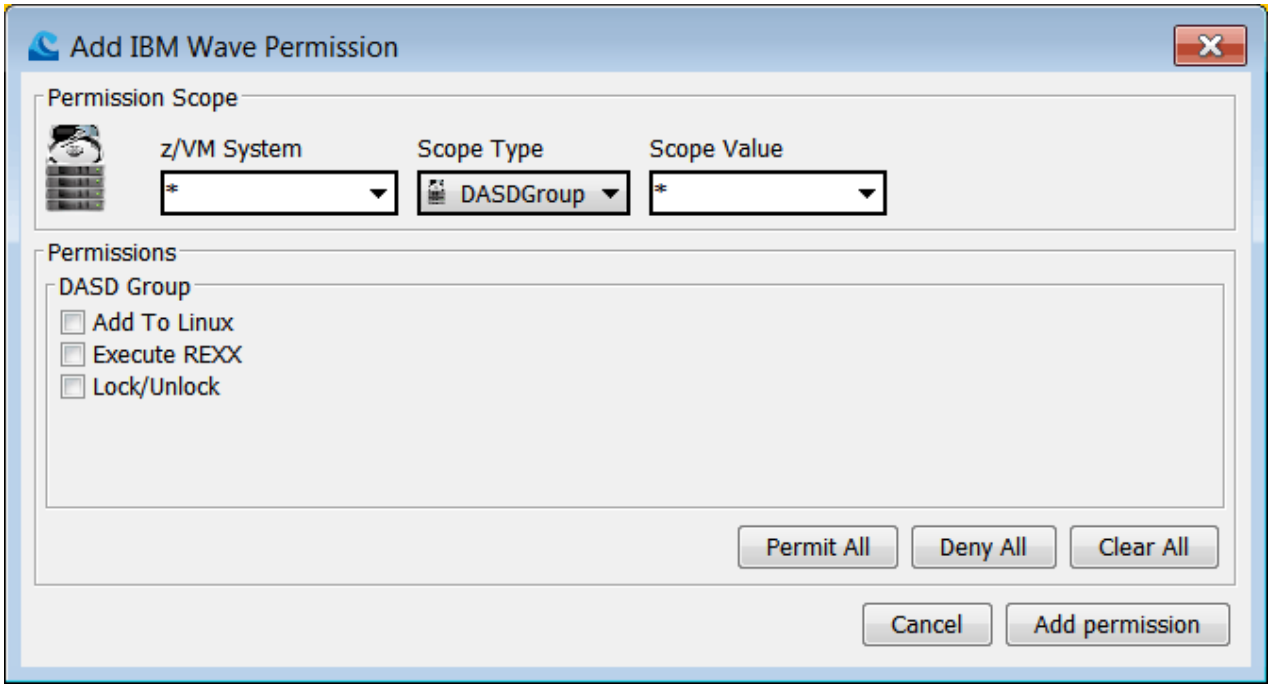


Figure 71. DASD Group Permissions

Device Pool Permissions

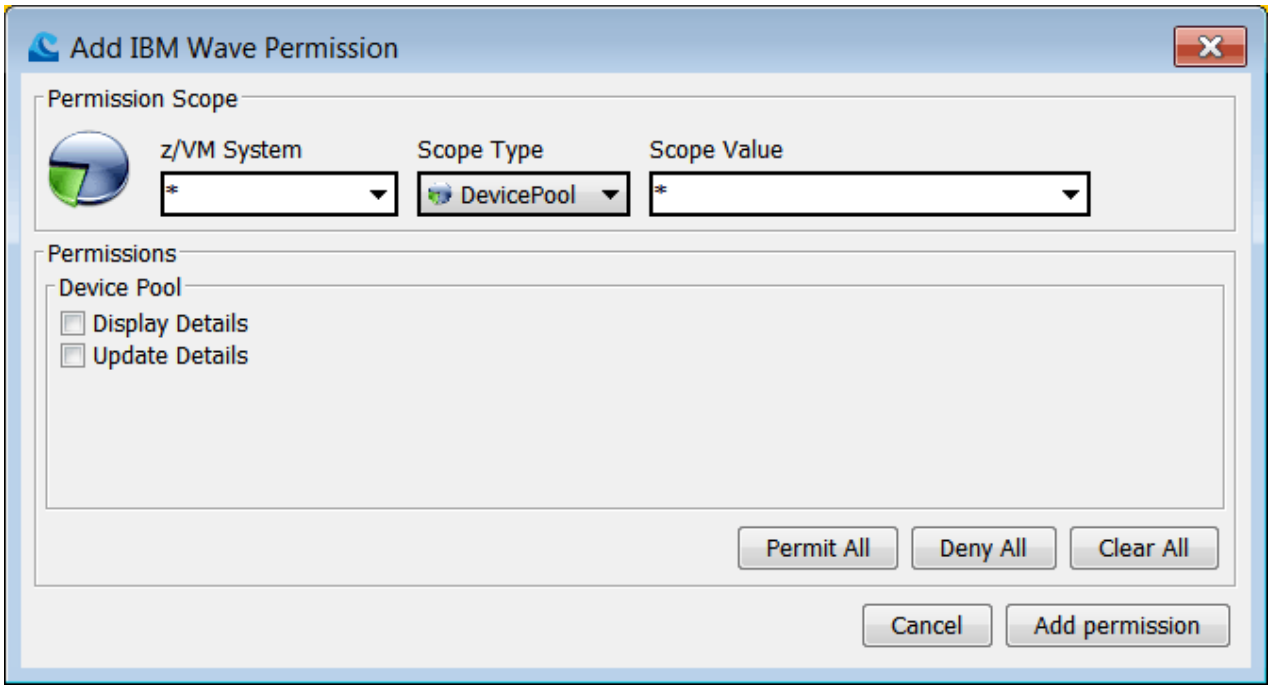


Figure 72. Device Pool Permissions

Copy Scopes and Permissions

The **Copy Permissions and Scope** function allows you to copy the scopes and permissions from another user.

In the **IBM Wave User Manager** > **Scopes and Permissions table**, you can **Copy Permissions and Scope** to a new user. Select the user row in the **Scopes and Permissions table**, and then right-click to **Copy Permissions and Scope**.

When you choose an IBM Wave user from the table, IBM Wave displays the user's scopes and permissions. With administrator privileges, you can change the scopes and permissions to copy to the edited IBM Wave User.

Creating and updating IBM Wave users

This topic explains how to create and update users in IBM Wave for z/VM.

To create and update IBM Wave users, use the **IBM Wave User Manager**. From the **IBM Wave Main Menu**, click **Administrative** > **Manage IBM Wave Users**.

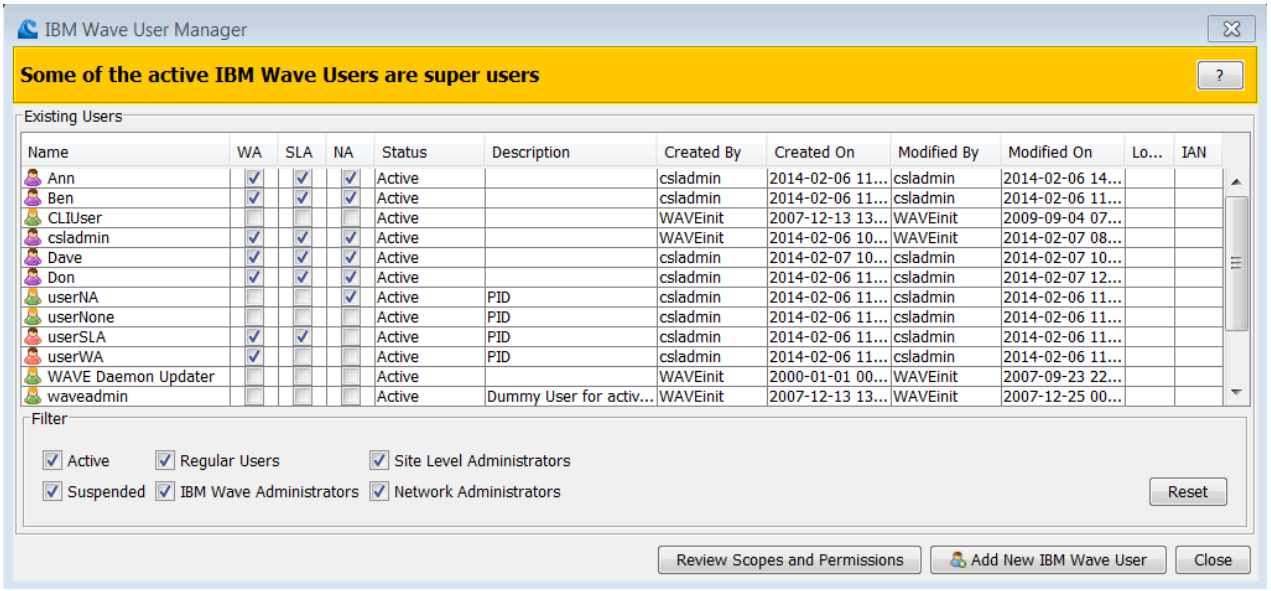


Figure 73. IBM Wave User Manager

An IBM Wave administrator can add and update IBM Wave users. Several industry standards prevent the deletion of IBM Wave users. The following fields are available in **Manage IBM Wave Users**:

- **Name** - The IBM Wave user name.
  - Regular users have a green user icon.
  - Administrator-type users have a red icon.
  - Super users have a purple icon.
  - LDAP users have a yellow icon.

  Note: IBM Wave users who use Active Directory authentication appear as `user_name@domain_name`.
- **WA** - Indicates that the user is an administrator-type IBM Wave user.
- **SLA** - Indicates that the user has the Site Level Administrator role.
- **NWA** - Indicates that the user has the Network Administrator role.
- **Status** - Indicates the status of the IBM Wave user as:
  - **Active** - The IBM Wave user is active and can log on to the application. If a user who was suspended due to inactivity is changed from **Suspend** to **Active** by the administrator, the user needs to log on before the next run of the **Update IBM Wave User Status** scheduled task.
  - **Suspended** - The IBM Wave user is suspended because of a security violation (such as trying to log on too often with a wrong password) or not logging in for the suspend account threshold, which defaults to 60 days. For more information, see the **Suspend Account Threshold** description in “Security parameters” on page 125.
  - **ASuspend or A-Suspend** (Administrative Suspend) - The IBM Wave User was suspended by an administrator. The IBM Wave user cannot log on IBM Wave.
- **Description** - A description of the user that is free-format text (up to 255 characters).

- **Created By/On** - Indicates the name of the IBM Wave administrator-type user who created the user and the time stamp.
- **Modified By/On** - Indicates the IBM Wave administrator-type user who last modified the user and the time stamp.
- **Locked** - Specifies whether the user is locked.
- **IAN** - Specifies whether the user has an Intelligent Active Note (IAN) attached. If so, you can hover over the IAN to read the contents.

You can sort and filter the table. To sort the table, click the title of the column you want to sort. Use the filter pane at the bottom of the window to hide an IBM Wave user-types. The filter performs an "AND" operation between all check boxes. To reset the filter, click **Reset**.

Note: If one or more active Super Users are defined in the system, a warning message displays at the top of the **IBM Wave User Manager** window. When you click the question mark "**?**" on the message, the message indicates the number of super users, and how to identify them in the **IBM Wave User Manager**.

Complete the following steps to create a new user.

1. Click the **Add New IBM Wave User** button in the **IBM Wave User Manager**. The **Create New IBM Wave User** opens as shown in Figure 74 on page 152.

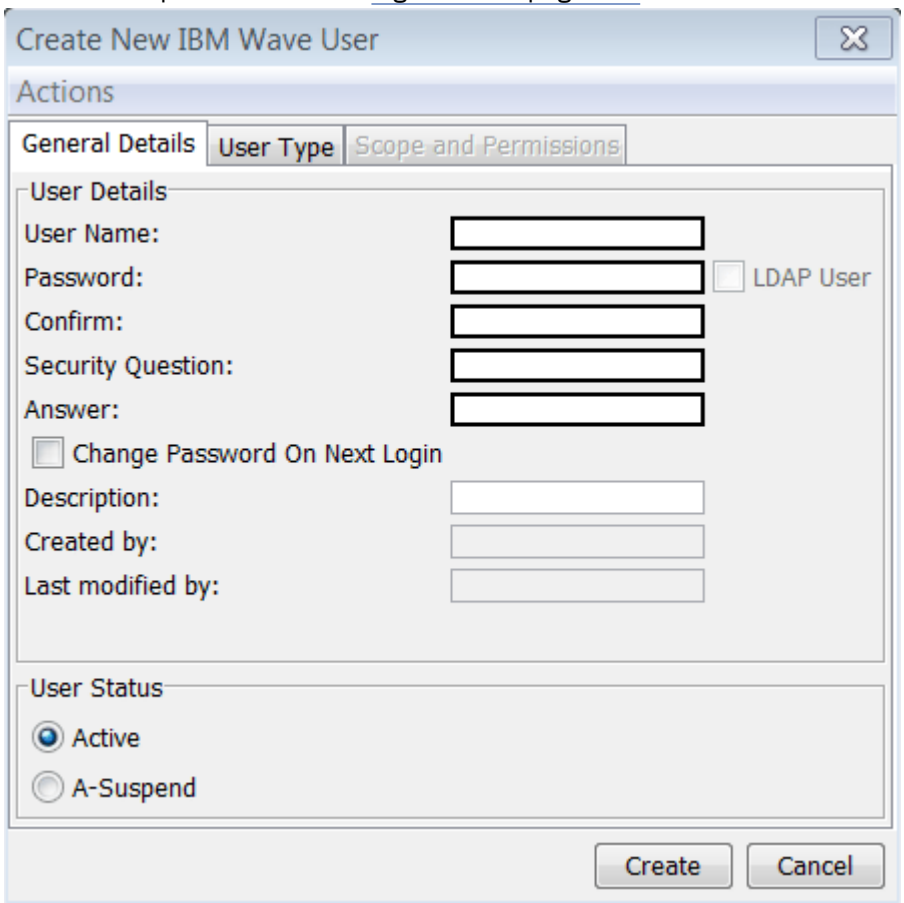


Figure 74. Create New IBM Wave User

2. Define the new user in the "General Details" tab, which contains the following fields:
   - **User Name** - The user's name.
   - **Password/Confirm** - The IBM Wave user password. The password must be at least 8 characters in length.
   - **Security Question/Answer** - A password reminder if the user forgets the password.
   - **Change Password on Next Login** - Hide the password from the IBM Wave administrator.
   - **Description** - The description for the user (up to 255 characters).

- **Created By** - The user who created the user.
- **Last Modified By** - The user who last modified the user.
- **User Status** - The status of the user.

3. Define the type of user and the user role in the "User Type" tab.
   - **Regular User or Administrator** - Define the type of the IBM Wave user.
   - **Network Admin, Site Level Admin, or both** - Define the role, if any, of the IBM Wave user.

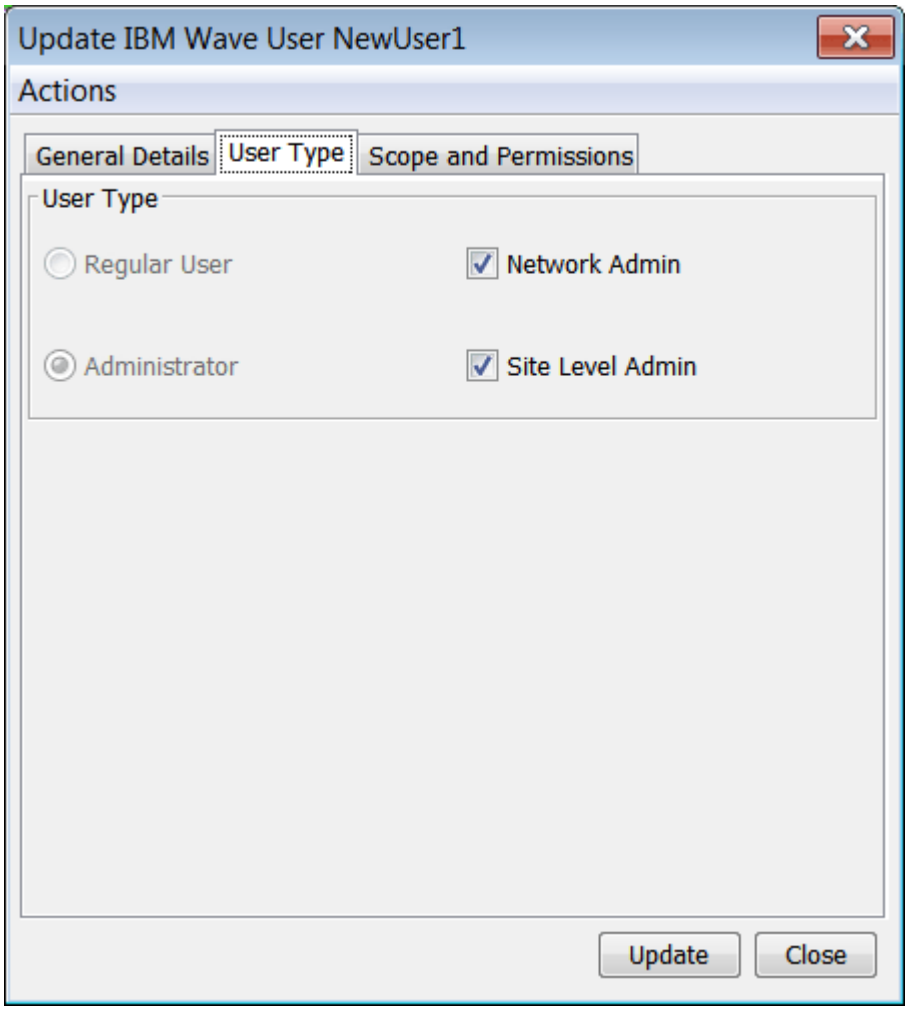


Figure 75. User Type tab

4. Define the resources that the user can access in "Scope and Permissions" tab.

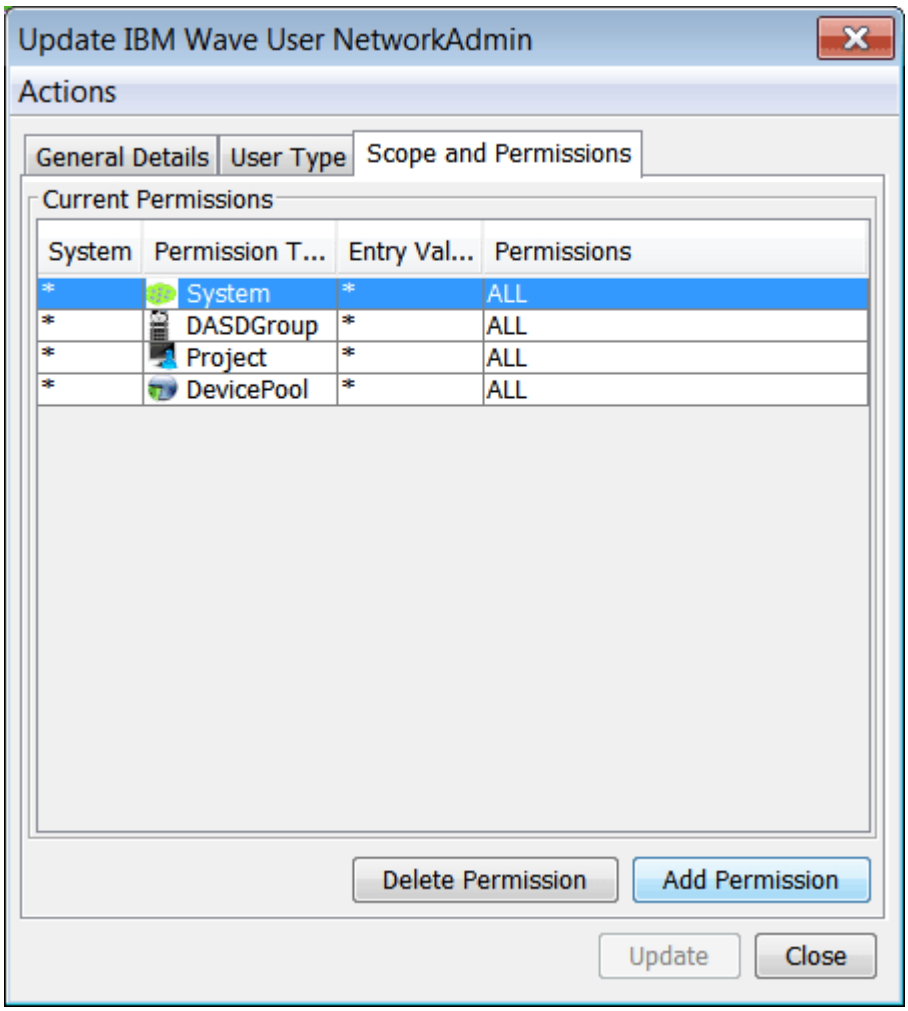


Figure 76. Scopes and Permissions tab

The "Scope and Permissions" tab contains the following fields.

- **System** - The z/VM LPAR for which the scope relates.
- **Permission Type** - The type of object to which the scope relates.
- **Entry Value** - The name of the object in the scope entry.
- **Permissions** - The actions allowed for the scope entry.

For more information about users, roles, scopes, and permissions, see Chapter 7, “User management,” on page 147.

Deleting IBM Wave Users

The Delete IBM Wave users function allows you to permanently delete user IDs from IBM Wave.

To Delete IBM Wave users in the IBM Wave User Manager, select the users in the existing users table. Right-click and select the **Delete Wave Users action**.

Note: The delete action is irreversible and causes all data related to the user to be deleted.

All log and audit records that pertain to the user or actions the user took in the past are unaffected by the delete action. Also, an audit message is created for each user that is deleted.

The delete action opens a multiple task window with a list of the users you want to delete.

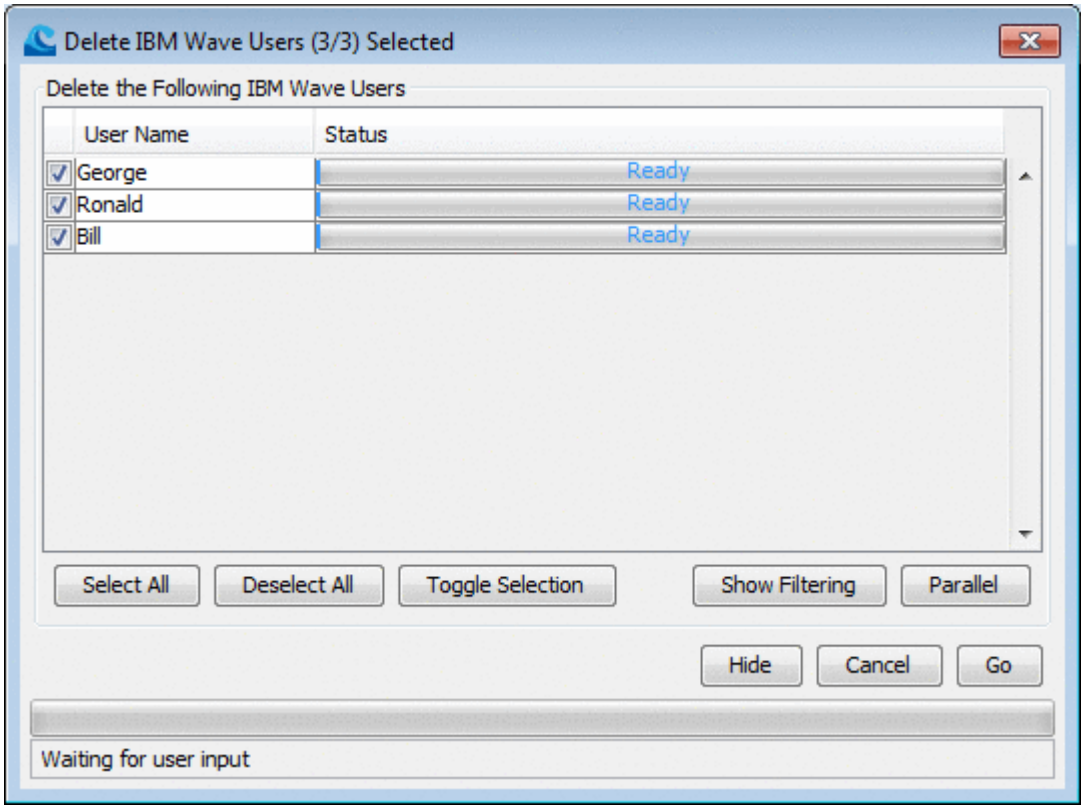


Figure 77. Delete IBM Wave Users

Click **Go** to send a work unit to the BTS that deletes all the checked wave users.

Only Site Level Administrators can delete IBM Wave users.
The last IBM Wave user who is a database user and a Site Level Administrator cannot be deleted.

Deletion of the IBM Wave internal users is not permitted.

Creating and updating IBM Wave user profiles

You can use the **Manage IBM Wave User Profiles** action to classify certain users and grant scopes and permissions on a generic basis.

Using LDAP integration, you can also associate LDAP groups to **IBM Wave User Profiles**. The users' scope and permission sets are based on the LDAP groups to which the user belongs.

If the scope and permissions are changed for a User Profile while the user is logged in, the updated scope and permissions do not take affect until the user logs out, and then back in.

To launch the **IBM Wave User Profile Manager**, from the IBM Wave main menu, select **Administrative** > **Manage IBM Wave User Profiles**.

You can view the user profiles by using the IBM Wave User Profile name or by the LDAP group to which the user profile is associated.

The fields in the **IBM Wave User Profile** table are similar to the fields in the **IBM Wave User Manager** table. The windows and fields for the **Add** and **Update** actions are similar to the actions for creating, and updating IBM Wave users.

In the **IBM Wave User Profile Manager**, as shown in Figure 78 on page 156, the key difference between the interfaces is the ability to associate a user profile with one or more LDAP groups.

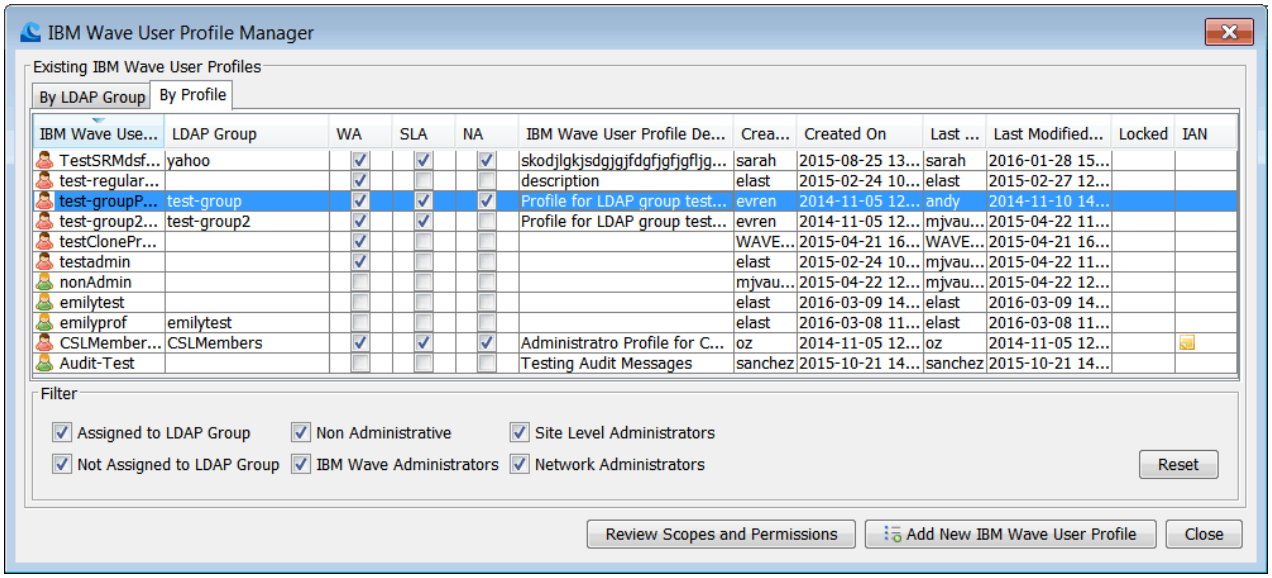


Figure 78. Create and Update Profiles

To associate an LDAP group with an IBM Wave Profile, click **Add Group** and then provide the LDAP group's name. To remove the association, select the group from the table, and click **Delete Group**.

IBM Wave User Permissions Cleaner

In the **Wave User Manager**, click **Review Scopes and Permissions** to open the **IBM Wave User Permissions Cleaner** window. The **IBM Wave User Permissions Cleaner** contains all of the scopes and permission definitions for the system, and looks similar to Figure 79 on page 156.

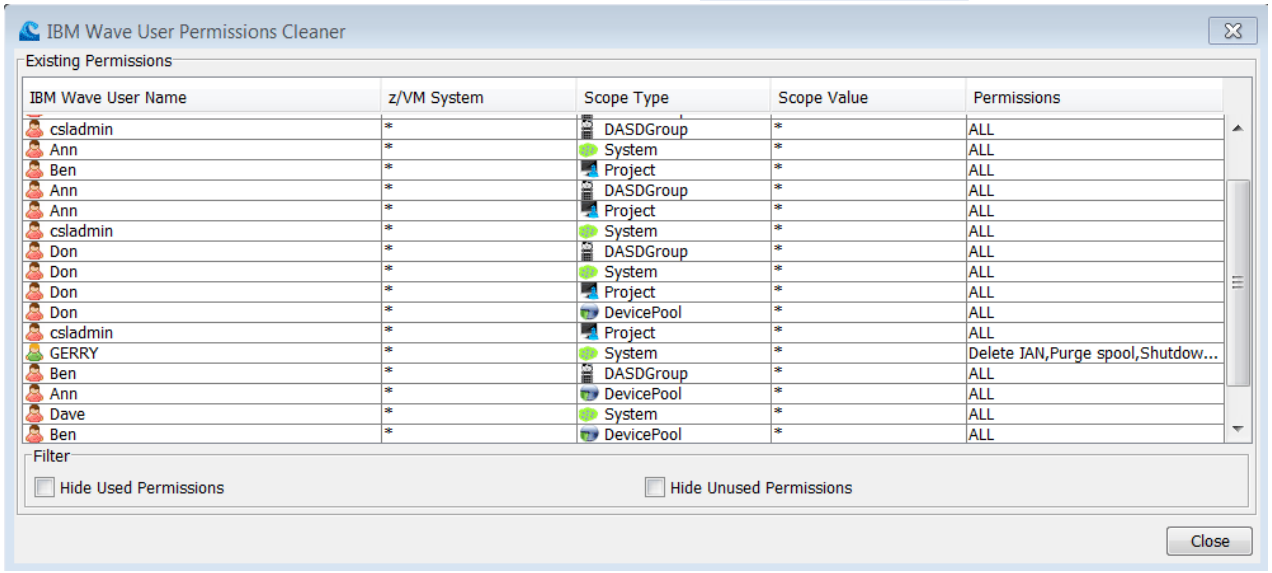


Figure 79. IBM Wave User Permissions Cleaner

The **IBM Wave User Permissions Cleaner** window displays "Scope Type", "Scope Value", and "Permissions" entries for all users who are defined as "Active". The **IBM Wave User Permissions Cleaner** does not display "Suspended" and "Administrator Suspended" (ASuspend) users.

Use the **IBM Wave User Permissions Cleaner** window to:

- View scopes and permissions values for users by using the sortable table columns. You can get a clear understanding of the user's scope for objects DASD Group, Device Pools, Projects, and z/VM Systems.

- Delete Scopes and Permissions in a mass. It is possible to select one or more Scope and Permission entries, right click, and select the "Delete Scope and Permission" action. This action deletes all of the selected entries.

  Note: IBM Wave prohibits the modification of a user who is logged-in user. When you try entries to delete that belong to a logged-in user, they are not deleted and a warning or error message will be displayed, detailing the entries which were not deleted. The same applies to IBM Wave Users which are locked.

Chapter 8. Audit Log Reporting feature

The IBM Wave Audit Log Reporting Feature tracks auditable events that are generated by IBM Wave and z/VM system actions.

IBM Wave writes an audit record for each event or action that occurs within IBM Wave. The events that you can audit include actions from z/VM systems, managed guests, Secure Shell (SSH) and application programming interfaces (API). You can work with the audit files to filter, export, and truncate records according to the auditing standards for your installation.

The following topics cover the feature of the Audit Log Reporting Feature:

- To select the audit events that you want to track, and to understand how audit data is stored, and see “Audit Log parameters” on page 129.
- For the **Audit Log Preview Options**, see “GUI parameters” on page 116.
- For instructions about working with the **IBM Wave Audit Log Display**, see “Displaying audit log events” on page 159.
- For an overview of the IBM Wave message format, see “IBM Wave message format” on page 203.
- For a complete list of messages issued by IBM Wave for audit events, see Appendix N, “IBM Wave messages,” on page 203.

Displaying audit log events

To view the **IBM Wave Audit Log Display**, from the **IBM Wave main menu**, click **User Tasks** > **View Audit Log**.

Before you begin

To view the Audit Log parameters from the **IBM Wave Main Menu**, go to **Administrative** > **Manage Parameters** and click on the **Audit Log** tab. Make the following decisions about the auditable events that are tracked for your installation:

- Do you want IBM Wave to write the audit log records to SYSLOG?
- How much audit log data does your installation want to maintain on IBM Wave after the audit log is truncated?
- What kind of manageable events do you want IBM Wave to record in the audit log events?
- Do you want the work unit start and end messages to appear in the audit log events?

Use your answers to adjust the "Manageable Audit Log Event Types" parameters in the **Audit Log** tab. For more information, see “Audit Log parameters” on page 129.

About this task

The **IBM Wave Audit Log Display** ***(Total number of audit log records)*** window can display 25,000 audit messages that IBM Wave issues. You can filter the table to display all of the data or the exact data that you require such as User, System, Token, or all. Both of the **Date/Time** fields are required.

For example, you can filter the **IBM Wave Audit Log Display** table to view all users ("*") on any system that begins with "VM*" as shown in Figure 80 on page 160.

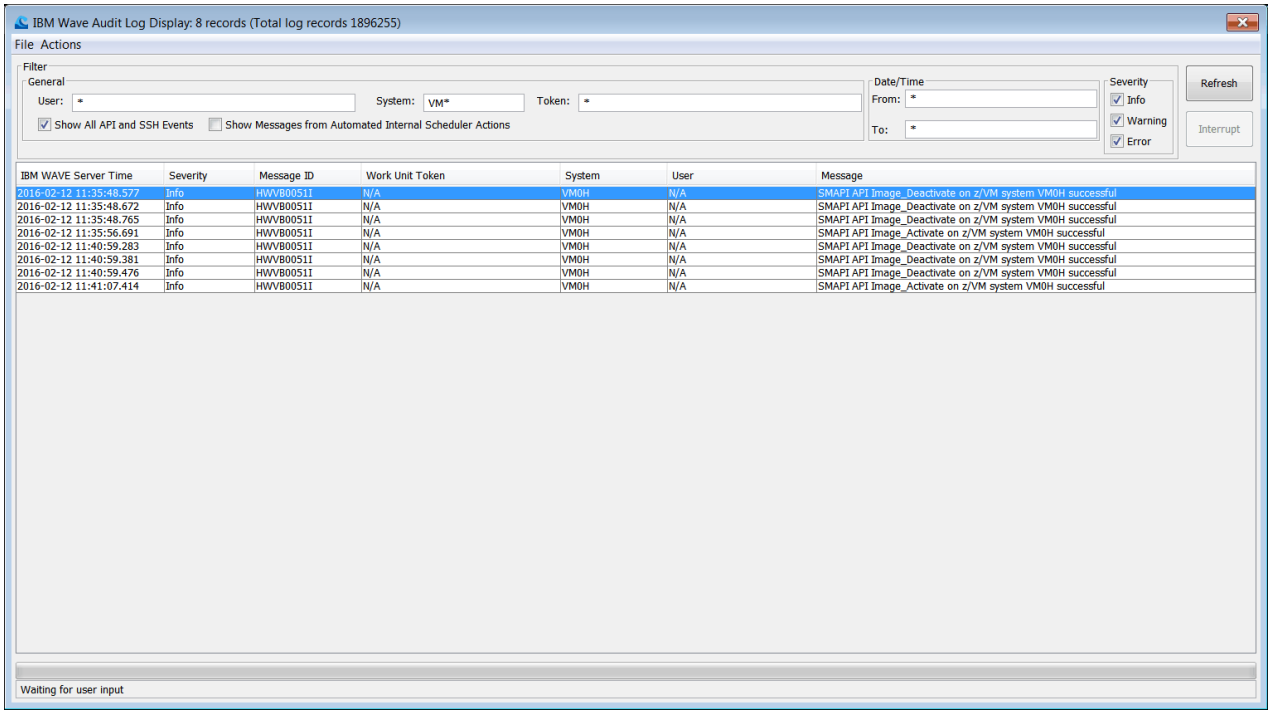


Figure 80. Audit Log Display

Note: IBM Wave Site Level Administrator users, with all inclusive scope, can view all audit messages that were generated by the system. Other user types can view audit messages that they generated by their own user.

Procedure

1. To open the **IBM Wave Audit Log Display**, from the **IBM Wave Main Menu**, click **User Tasks** > **View Audit Log**.

   The **IBM Wave Audit Log Display** opens.
2. To populate the **IBM Wave Audit Log Display** table, as shown in Figure 80 on page 160, select the filter options, and then click **Refresh** (or press **Enter)**. To stop or change the filter options before all of the log records load, click **Interrupt**.

   To filter the Audit Log data, you must select one of the following **Filter** options.

   General

   User: Specify a user, user group, task, or scheduled task. You can use the asterisks (star) character to retrieve a defined pattern of all possible users or scheduled tasks within a possible string. For example:

   - Specify "a*" to return all user IDs that begin with either "a" or "A".
   - Specify "Wave*" to return all scheduled tasks that begin with "Wave", such as the `WAVEBTSScheduler`.

   System: Specify a system name to see only the audit events that are issued on the specified system. Or use the asterisks (star) character to retrieve a defined pattern of all possible systems within the string.

   For example, specifying VM* returns the events that occurred on a system name that begins with the characters "VM" or "vm". You can also filter and view events that apply only to IBM Wave, such as Secure Shell (SSH) and APIs by specifying "N/A" in the **System** field.

   Token: Specify the work unit token, which is the unique identifier that distinguishes and connects the user to one or more related actions.

Show All z/VM API and SSH Events
Select this check box to display all z/VM API (SMAPI and IBM Wave Service Machines) and SSH events in the **IBM Wave Audit Log Display** table.

Show Messages from Automated Internal Scheduler Actions
Select this check box to display the messages from the automated internal scheduler in the **IBM Wave Audit Log Display** table.

Date/Time
The Date/Time fields must have a value or an asterisk ("*") to populate the **IBM Wave Audit Log Display** table. You can use an asterisk in one or both fields to indicate the start or end time. Select "From" for the start date and "To" for the end date. IBM Wave provides a date and time option. After you select the values you want, click **OK**. IBM Wave uses 24-hour notation. The data collection time is 00:00:00 to 23:59:59.

Severity
The message severity.

3. When you are finished populating the **IBM Wave Audit Log Display** table, you can sort the data by using the table headings.

   The **IBM Wave Audit Log Display** contains the following table headings:

   IBM Wave Server Time
   The date and time when the event occurred.

   Severity
   The severity of the message, which can be Informational or Error.

   Message ID
   The message ID. For IBM Wave messages, see “HWVA0001E” on page 203. For details about the IBM Wave message format, see “IBM Wave message format” on page 203.

   Work Unit Token
   Each action that occurs within IBM Wave is assigned a work unit token ID. The work unit token IDs are generated and assigned to messages that result from a single Wave action and allow events to be more easily connected to one another.

   - "N/A": Messages that are not associated with a work unit.
   - "-": Messages that were produced before the Audit Logging feature was added to IBM Wave.

   System
   The z/VM system or N/A for IBM Wave events.

   User
   The user ID or the scheduled task.

   Message
   The message that IBM Wave issues. To control what messages IBM Wave issues, adjust the options in the “Audit Log parameters” on page 129.

   Reason
   The reason and return code, if available.

4. You can optionally save the messages that are shown in the **IBM Wave Audit Log Display** as a comma-separated value (.csv) file. From the **IBM Wave Audit Log Display** main menu, select **File > Save Log As**.
5. You can optionally truncate the log. From the **IBM Wave Audit Log Display** main menu, select **Actions > Truncate Log**.

   Note: To keep the file system size manageable, IBM Wave automatically truncates the log data every 24 hours and stores the file, `IBM-Wave-Log`*`YYYYDDD-HH:MM:SS`*, in the `tmp` file directory.

Results
You are now displaying audit log events and can maintain log records for all audit events that happened in IBM Wave.

What to do next

You are done, but if you need additional information, see the following topics:

- To tailor the auditable event types and truncate the number of **Audit Log Display** messages, see "Audit Log parameters" on page 129.
- To adjust the number of **Audit Log Preview** messages, see "GUI parameters" on page 116.
- To review the overall message format and auditable events messages (HWV prefix), see "IBM Wave message format" on page 203 and Appendix N, "IBM Wave messages," on page 203.

Chapter 9. Uninstalling IBM Wave

IBM Wave is as non-intrusive as possible. Use the following procedure to uninstall IBM Wave for z/VM.

Tasks for uninstalling IBM Wave

Use the following procedure to uninstall IBM Wave.

About this task

The following tasks are required when you uninstall IBM Wave.

Procedure

1. Delete the WAVESRV by using one of the following two methods.
   a) Delete the entire WAVESRV Virtual Server, or servers, depending on your site's installation.
   b) Use the following commands to remove the IBM Wave installation from the WAVESRV virtual server or servers. This step removes the IBM Wave database and all relevant files and configurations.
      - To get the package name, enter:

        ```
        rpm – qa | grep
        ```

      - To remove the installation, enter:

        ```
        rpm -e package_name
        ```

2. Delete the IBM Wave service machines from every z/VM System (LPAR) that was managed by IBM Wave.
3. Remove entries for the IBM Wave service machines from your enterprise security manager (ESM).
4. Delete the dummy DIRMAINT DASD region on every z/VM System (LPAR) running DIRMAINT that was managed by IBM Wave.

What to do next

Some IBM Wave actions leave a remark in some configuration files. For example, creating z/VM Guests with IBM Wave leaves a remark in the z/VM Guest's directory entry, which signifies that the guest was created by using IBM Wave for z/VM. You can remove the remarks.

Appendix A. Linux distribution support

Managed guests Linux OS distribution release support

The following Linux operating system (OS) distributions are fully supported:

- Red Hat Enterprise Linux 5 (RHEL 5)
- RHEL 6
- RHEL 7
- RHEL 8
- SUSE Linux Enterprise Server 11 (SLES 11)
- SLES 12
- SLES 15
- Ubuntu Server 16

Note: Different Wave actions cause different Linux commands to run on the selected managed guests (the **cat** command, for example). These commands are searched in the following directories only, in the following order:

- `/usr/local/sbin`
- `/usr/local/bin`
- `/usr/sbin`
- `/usr/bin`
- `/sbin`
- `/bin`

Appendix B. A sample .csv file for importing guest attributes

The following example shows a .csv file that you can use to import guest attributes. This example assumes that there are two custom attributes defined: `Application` and `Functionality`, each with the possible values preconfigured. Site Defined Groups SDG1, SDG2, and SDG3 also exist.

```
----------------- Start of CSV File -----------------------
intr_username,intr_system,intr_project,intr_SDG,attr_Application,
attr_Functionality,attr_Importance
DEMO100,CSLZVM,Proj1,SDG1,App1,Func1,High
DEMO101,CSLZVM,Proj2,SDG2,App2,Func2,Medium
DEMO102,CSLZVM,Proj3,SDG3,App3,Func3,Low
------------------ End of CSV File -----------------------
```

Appendix C. A sample WAVESRV directory entry

Figure 81 on page 169 is an example of a directory entry for the Linux (WAVESRV) virtual server.

```
*****************************************************************
USER WAVESRV masked_password 2G 3G GC
 CPU 00
   IPL CMS
   MACHINE ESA 4
   OPTION QUICKDSP
   CONSOLE 0009 3215
   NICDEF 0800 TYPE QDIO LAN SYSTEM GLAN/VSWITCH_name
   SPOOL 000C 3505 A
   SPOOL 000D 3525 A
   SPOOL 000E 1403 A
   LINK MAINT 0190 0190 RR
   LINK MAINT 019D 019D RR
   LINK MAINT 019E 019E RR
   MDISK 0191 3390 1 3 DASD_volume_name - This is a CMS minidisk
   MDISK 0150 3390 4 3000 DASD_volume_name - This minidisk will be used for ,,/''
   MDISK 0151 3390 3004 4500 DASD_volume_name - This minidisk will be used for ,,/var''
   MDISK 0152 3390 7504 2500 DASD_volume_name - This minidisk will be used for swap
*****************************************************************
```

Figure 81. A sample directory entry for WAVESRV

Appendix D. Changing the IBM Wave server IP address or host name

Use the following procedure to change the IP address or host name of your Wave server or begin using a host name for the **Launch IBM Wave** button. This will update the `.jnlp` files that are associated with the button.

About this task

Use the following procedure to change the IP address or host name for your Wave server.

Procedure

1. Update the IP address or host name of the Wave server according to your Linux distribution instructions.
2. SSH to the Wave server and run the following as `root` to update the IP address or host name in the Wave `.jnlp` files:

   ```
   /usr/wave/install/wavesrv-name.sh update
   ```

3. Make sure both applications load properly. Open a web browser and enter the new IP address or host name. Test the **Launch IBM Wave** button.

Results

When the IBM Wave application loads properly, you have changed the IP address or host name successfully.

Appendix E. Shared directory considerations for service machines

By default, IBM Wave requires the IBM Wave service machines to be named differently on z/VM systems that share a directory. When adding a z/VM system to IBM Wave management, it is possible to specify the names of the three service machines in the bottom portion of the window.

IBM Wave supports the use of the internal form of the SYSAFFIN directory statement for the service machines. However, the implementation must be done manually by a z/VM administrator. Both minidisks for the short service machine (WAVEWRKS by default) must be defined for each z/VM system. Because the other service machine simply links to the short service machine's minidisks, the configuration can be left "as-is" without the SYSAFFIN statement (prefix or internal).

Define the IBM Wave service machines

Use the following directions to define IBM Wave service machines.

1. Define the user ID for the short, WAVEWRKS, service machine:
   a. Log on as the MAINT user ID.
   b. Create a file called WAVEWRKS DIRECT on the A minidisk:

```
"X WAVEWRKS DIRECT A"
```

   c. Paste the following content into the file:

```
USER WAVEWRKS PASSWORD 128M 512M ABCDEFG
         IPL CMS
         MACHINE ESA 4
         OPTION LNKNOPAS DIAG88
         CONSOLE 0009 3215
         SPOOL 000C 3505 A
         SPOOL 000D 3525 A
         SPOOL 000E 1403 A
         LINK MAINT 0190 0190 RR
         LINK MAINT 019D 019D RR
         LINK MAINT 019E 019E RR
         LINK TCPIP 0592 0592 RR
         LINK MAINT 0193 0193 RR
         *CSLTAG01: WAVE-INTERNAL
```

   Where *PASSWORD* is a password that you chose for the three service machines.
   d. For each z/VM system sharing the directory, add a SYSAFFIN statement block as follows, where SYSTEM1, SYSTEM2, and SYSTEM3 are replaced with the z/VM systems that share the directory:

```
SYSAFFIN SYSTEM1
AMD 0191 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI
AMD 0399 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI

SYSAFFIN SYSTEM2
AMD 0191 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI
AMD 0399 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI

SYSAFFIN SYSTEM3
AMD 0191 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI
AMD 0399 3390 AUTOG 100 TEST RR PW
READ WRITE MULTI
```

where:

TEST must be the name of the storage group to be used.
READ is the read password chosen for the service machine minidisks.
WRITE is the write password chosen for the service machine minidisks.
MULTI is the multiple password chosen for the service machine minidisks.

e. Save and exit from XEDIT and then enter the following command at the CMS Ready prompt to create the z/VM user:

```
"DIRM ADD WAVEWRKS"
```

2. Create the long and CSC service machines:

a. Log on as the MAINT user.

b. Create a file called WAVEWRKL DIRECT on the A minidisk and copy the following example into it:

```
USER WAVEWRKL <PASSWORD> 128M 512M ABCDEFG
         IPL CMS
         MACHINE ESA 4
         OPTION LNKNOPAS DIAG88
         CONSOLE 0009 3215
         SPOOL 000C 3505 A
         SPOOL 000D 3525 A
         SPOOL 000E 1403 A
         LINK MAINT 0190 0190 RR
         LINK MAINT 019D 019D RR
         LINK MAINT 019E 019E RR
         LINK TCPIP 0592 0592 RR
         LINK MAINT 0193 0193 RR
         LINK WAVEWRKS 0191 0191 RR
         LINK WAVEWRKS 0399 0399 RR
         *CSLTAG01: WAVE-INTERNAL
```

Where *<PASSWORD>* is a password, chosen by you, for the three service machines.

c. Create the z/VM user:

```
"DIRM ADD WAVEWRKL"
```

d. Create a file that is named `WAVEWRKC DIRECT` on minidisk A and copy the following example into it:

```
USER WAVEWRKC <PASSWORD> 128M 512M ABCDEFG
         IPL CMS
         MACHINE ESA 4
         OPTION LNKNOPAS LNKE DIAG88
         CONSOLE 0009 3215
         SPOOL 000C 3505 A
         SPOOL 000D 3525 A
         SPOOL 000E 1403 A
         LINK MAINT 0190 0190 RR
         LINK MAINT 019D 019D RR
         LINK MAINT 019E 019E RR
         LINK TCPIP 0592 0592 RR
         LINK MAINT 0193 0193 RR
         LINK WAVEWRKS 0191 0191 RR
         LINK WAVEWRKS 0399 0399 RR
         *CSLTAG01: WAVE-INTERNAL
```

Where *<PASSWORD>* is a password chosen by you for the three service machines.

e. Add the user:

```
DIRM ADD WAVEWRKC
```

3. Format the 191 and 399 minidisks for CMS usage. The format must be done in every system.
   a. Log in into the MAINT user and make sure that WAVEWRKS, WAVEWRKL, and WAVEWRKC are not logged on to make sure to get the 191 and 399 disks with write access. If necessary, force logoff.
   b. Link to the 191 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 191 1191 WR"
```

   c. Format the minidisk:

```
"FORMAT 1191 J"
```

      When asked for a label, enter WAV191.
   d. Detach the disk:

```
"rel J (DET""
```

   e. Link to the 399 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 399 1399 WR"
```

   f. Format the minidisk:

```
"FORMAT 1399 J"
```

      When asked for a label, enter WAV399.
   g. Detach the disk:

```
"rel J (DET"
```

Repeat these steps for each system that is sharing directories.

Return to Chapter 2, “Installing and customizing IBM Wave,” on page 51 as needed.

Appendix F. Considerations for the service machines when working with SSI

By default, IBM Wave requires the IBM Wave service machines to be named differently on z/VM systems within a single system image (SSI) cluster. When adding a z/VM system to IBM Wave management, you can specify the three service machine names in the bottom portion of the window.

Define service machines within an SSI cluster

IBM Wave can support the definition of the service machines as identities with subconfigurations. This implementation, however, must be done manually by a z/VM administrator. Both minidisks of the short service machine (WAVEWRKS by default) must be defined for each z/VM System in a separate configuration. Because the other service machine simply links to the short Service Machine's minidisks, it can be left "as-is".

Use the following procedure to define the three service machines using identities and sub-configurations in an environment with two z/VM systems in the SSI cluster (SSI1 and SSI2):

Note: These directions are DirMaint-based.

1. Define the user for the short service machine (WAVEWRKS):
   a. Log in as the MAINT user.
   b. Create a file named `WAVEWRKS DIRECT` on the A minidisk:

```
IDENTITY WAVEWRKS <PASSWORD> 128M 512M ABCDEFG
        IPL CMS
        MACHINE ESA 4
        OPTION LNKNOPAS DIAG88
        CONSOLE 0009 3215
        SPOOL 000C 3505 A
        SPOOL 000D 3525 A
        SPOOL 000E 1403 A
        LINK MAINT 0190 0190 RR
        LINK MAINT 019D 019D RR
        LINK MAINT 019E 019E RR
        LINK TCPIP 0592 0592 RR
        LINK MAINT 0193 0193 RR

Where <PASSWORD> is a password chosen by you for the three
service machines.
```

   c. Add the identity: `DIRM ADD WAVEWRKS`
2. Define the subconfigurations for each system in the SSI cluster. In the example, two z/VM systems are in the cluster SSI1 and SSI2, and therefore two subconfigurations are needed. Perform the steps for each subconfiguration, replacing the variables with the appropriate values:
   a. Log in as the MAINT guest.
   b. Create a file named *subconfig_name* DIRECT on the A minidisk. For example:

```
X WAVEWRKS DIRECT A
```

   c. Paste the following content into the file, and then save it and exit:

```
SUBCONFIG<SUBCONFIG_Name>
*CSLTAG01: WAVE-INTERNAL
AMD 0191 3390 AUTOG 100 <TEST> RR PW
<READ> <WRITE> <MULTI>
AMD 0399 3390 AUTOG 100 <TEST> RR PW
<READ> <WRITE> <MULTI>
```

   where:

<TEST> is the name of the storage group to be used.
<READ> is the read password chosen for the service machine minidisks.
<WRITE> is the write password chosen for the service machine minidisks.
<MULTI> is the multi password chosen for the service machine minidisks.

d. Add the sub-configuration:

```
"DIRM ADD <SUBCONFIG_Name> BUILD ON <System Name> IN WAVEWRKS"
```

e. Repeat steps a - d for any additional SUBCONFIG statements and replace:

- Where <*SUBCONFIG Name*> is a name for the SUBCONFIG. For example, WAVWRKS1 for System 1, WAVWRKS2 on System 2.
- Where <*System Name*> is the z/VM System name.

3. Create the long and CSC service machines IDENTITIES.

a. Login to the MAINT user and ensure that WAVEWRKS, WAVEWRKL, and WAVEWRKC are not active. Force logout if necessary to ensure the 191 and 399 disks have write access.

b. Create a file named `WAVEWRKL DIRECT` on disk A.

c. Copy the following information into the file:

```
IDENTITY WAVEWRKL <PASSWORD> 128M 512M ABCDEFG
       IPL CMS
       MACHINE ESA 4
       OPTION LNKNOPAS DIAG88
       CONSOLE 0009 3215
       SPOOL 000C 3505 A
       SPOOL 000D 3525 A
       SPOOL 000E 1403 A
       *CSLTAG01: WAVE-INTERNAL
       LINK MAINT 0190 0190 RR
       LINK MAINT 019D 019D RR
       LINK MAINT 019E 019E RR
       LINK TCPIP 0592 0592 RR
       LINK MAINT 0193 0193 RR
       LINK WAVEWRKS 0191 0191 RR
       LINK WAVEWRKS 0399 0399 RR
```

Where <*PASSWORD*> is the password, chosen by you, for the three service machines.

d. Add the identity:

```
"DIRM ADD WAVEWRKL"
```

e. Create a file named `WAVEWRKC DIRECT` on the A minidisk. Copy the following information into the file:

```
IDENTITY WAVEWRKC <PASSWORD> 128M 512M ABCDEFG
       IPL CMS
       MACHINE ESA 4
       OPTION LNKNOPAS LNKE DIAG88
       CONSOLE 0009 3215
       SPOOL 000C 3505 A
       SPOOL 000D 3525 A
       SPOOL 000E 1403 A
       *CSLTAG01: WAVE-INTERNAL
       LINK MAINT 0190 0190 RR
       LINK MAINT 019D 019D RR
       LINK MAINT 019E 019E RR
       LINK TCPIP 0592 0592 RR
       LINK MAINT 0193 0193 RR
       LINK WAVEWRKS 0191 0191 RR
       LINK WAVEWRKS 0399 0399 RR
```

Where <*PASSWORD*> is a password, chosen by you, for the three service machines.

f. Add the IDENTITY:

```
"DIRM ADD WAVEWRKC"
```

4. Format the 191 and 399 minidisks for use by CMS. This must be done in every system in the SSI cluster. (In this example, the procedure has to be done twice: once for SSI1, and once for SSI2):
   a. Login as the MAINT user.
   b. Link to the 191 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 191 1191 WR"
```

   c. Format the minidisk:

```
"FORMAT 1191 J"
```

   d. When asked for a label, enter WAV191
   e. Detach the disk:

```
"rel J (DET"
```

   f. Link to the 399 minidisk of WAVEWRKS with WRITE permissions:

```
"LINK WAVEWRKS 399 1399 WR"
```

   g. Format the minidisk:

```
"FORMAT 1399 J"
```

   h. When asked for a label, enter WAV399
   i. Detach the disk:

```
"rel J (DET"
```

   j. Repeat steps a-i for each system in the SSI cluster.

Return to Chapter 2, “Installing and customizing IBM Wave,” on page 51 as needed.

Appendix G. Configuring VM: Secure

IBM Wave requires that the z/VM System Management API is configured and functioning. Therefore, you must follow the customization steps in Appendix H, “Customizing VM: Secure to use SMAPI,” on page 183.

Follow the instructions to configure the SMAPI to work properly with VM:Secure. In addition, you must update the VM:Secure `SECURITY CONFIG` file with an `ENABLE PWPHRASE` record.

Requirement: An Authorized API User is required.

VM: Secure requirements:

VM: Secure users must install:

1. Level 3.1 or later of VM: Secure.
2. Fix pack R067338 and individual fixes R067266, R070893, and R072116.

- If extents are not defined in subpools, they must be configured for IBM Wave to manage the storage. There must be only one extent per each DASD Volume. The extent name must match the DASD Volume name (VOLSER).
- If extents are defined in subpools, their names must match the name of the DASD Volume on which they are defined.

  Remember: IBM Wave does not support multiple extents that are defined on one DASD Volume.
- ACIGROUP named WAVEACIG must be pre-configured on the z/VM system with the following definitions:

  1. In the `CONFIG.SECURITY` file:

     ```
     GROUP WAVEACIG
     ```

  2. In the `SYSTEM RULES` file:

     ```
     ACCEPT WAVEACIG LINK * * (GROUP NOPASS
     ACCEPT WAVEACIG AUTOLOG (GROUP NOPASS
     ACCEPT WAVEWRKS LINK (HISTORY NOPASS
     ACCEPT WAVEWRKC LINK (HISTORY NOPASS
     ACCEPT WAVEWRKL LINK (HISTORY NOPASS
     ACCEPT WAVEWRKS AUTOLOG (HISTORY NOPASS
     ACCEPT WAVEWRKC AUTOLOG (HISTORY NOPASS
     ACCEPT WAVEWRKL AUTOLOG (HISTORY NOPASS

     ACCEPT WAVEWRKS VALIDATE
     ACCEPT WAVEWRKL VALIDATE
     ACCEPT WAVEWRKC VALIDATE
     ACCEPT WAVEWRKS DIAG88
     ACCEPT WAVEWRKC DIAG88
     ACCEPT WAVEWRKL DIAG88
     ACCEPT VSMPROXY DIAG88
     ACCEPT VSMREQI6 DIAG88
     ACCEPT VSMEVSRV DIAG88
     ACCEPT PERFSVM  DIAG88
     ```

     – The LINK rules permit the IBM Wave service machines to link to any minidisk without password verification.

     – The AUTOLOG rules permit the IBM Wave service machines to the autolog virtual machines without password verification.

     – The VALIDATE rules permit the IBM Wave service machines to use the password validation programming interfaces.

  - The DIAG88 rules permit the IBM Wave service machine to use code X'88' to validate user authorizations and link minidisks.

3. In the WAVEACIG GROUP RULES file:

```
ACCEPT * LINK 399 * (NOPASS
```

- The following entries must be present in the AUTHORIZ.CONFIG file:

```
LIST   *WAVEWRK WAVEWRKS WAVEWRKL WAVEWRKC
GRANT * TO VSMWORK1
GRANT * TO VSMWORK2
GRANT * TO VSMWORK3
GRANT * OVER *ALL TO WAVEWRKS
GRANT * OVER *ALL TO WAVEWRKC
GRANT * OVER *ALL TO WAVEWRKL
```

- The following entries must be present in the VMSECURE MANAGERS configuration file. In the example, the entries for *POOLX* and *POOLY* are DASD pools, which are specific to your environment. Be sure to make IBM Wave a manager for DASD pools that are defined for IBM Wave use.

```
MANAGER WAVEWRKS * POOLX POOLY
SKELETON WAVEWRKS GENERAL
DEVTYPE WAVEWRKS 3390

MANAGER WAVEWRKL * POOLX POOLY
SKELETON WAVEWRKL GENERAL
DEVTYPE WAVEWRKL 3390

MANAGER WAVEWRKC * POOLX POOLY
SKELETON WAVEWRKC GENERAL
DEVTYPE WAVEWRKC 3390
```

For more information, see *z/VM Secure Configuration Guide*.

TCP/IP

IBM Wave makes use of the z/VM FTP server to transfer files to minidisks owned by the IBM Wave service virtual machines. If you have not already done so, perform the VM:Secure customization steps for TCP/IP that are identified in *z/VM: TCP/IP Planning and Customization*.

z/VM: TCP/IP Planning and Customization contains the complete instructions for configuring TCP/IP to work with an external security manager.

For more information, see the *z/VM: Secure Configuration Guide*.

Service Machine

IBM Wave assumes that the VM:Secure code resides on minidisk 193 of the VM:Secure manager guest. IBM Wave service machines link to that minidisk as virtual address 293. If VM:Secure client code does not reside on that disk, modify the service machine directory entry and add a LINK statement that links to the correct VM:Secure code location.

For example, if the VM:Secure code resides on MAINT 325, add a LINK statement in the service machine's directory entry.

```
LINK MAINT 325 293 RR
```

Appendix H. Customizing VM: Secure to use SMAPI

When your installation is using the Systems Management API (SMAPI) support servers, some basic customization is needed for the VM: Secure configuration files. The following information is a detailed explanation of the customization:

1. The following statements must appear in the **SYSTEM RULES** configuration.

   General rules for the system:

   ```
   ACCEPT * LOGON
   ```

   Rules to enable FTP to the VM:

   ```
   ACCEPT FTP machine LINK * * (NOPASS
   ACCEPT FTP machine DIAG88
   ACCEPT FTP machine DIAGD4
   ACCEPT FTP machine SPOOL
   ```

   Rules for SMAPI machines:

   ```
   ACCEPT VSMWORK1 AUTOLOG (NOPASS
   ACCEPT GSMAPI TAG * (GROUP
   ```

   GSMAPI is a security group that is defined in the `CONFIG SECURITY` file. The group contains:

   - VSMWORK1
   - VSMWORK2
   - VSMWORK3
   - VSMREQIN
   - VSMREQIU
   - ACCEPT GSMAPI SPOOL (GROUP
   - ACCEPT GSMAPI DIAGD4 (GROUP
   - ACCEPT GSMAPI DIAG88 (GROUP

   Rules for SFS machines:

   ```
   ACCEPT VMSERVR LINK ** (NOPASS
   ACCEPT VMSERVU LINK ** (NOPASS
   ACCEPT VMSERVS LINK ** (NOPASS
   ```

2. The following statements must appear in the **MAINT USER RULES** configuration:

   ```
   ACCEPT GSMAPI LINK 190 RR (GROUP NOPASS
   ACCEPT GSMAPI LINK 19E RR (GROUP NOPASS
   ACCEPT GSMAPI LINK 193 RR (GROUP NOPASS
   ACCEPT VSMWORK1 LINK CF1 MR
   ACCEPT VSMWORK1 LINK CF2 MR
   ACCEPT VSMWORK2 LINK CF1 MR
   ACCEPT VSMWORK2 LINK CF2 MR
   ACCEPT VSMWORK3 LINK CF1 MR
   ACCEPT VSMWORK3 LINK CF2 MR
   ACCEPT * LINK 190 RR
   ACCEPT * LINK 19D RR
   ACCEPT * LINK 19E RR
   ```

3. The following statements must appear in the **TCPMAIN USER RULES** RULES configuration:

   ```
   ACCEPT GSMAPI LINK 0591 RR (GROUP NOPASS
   ACCEPT GSMAPI LINK 0592 RR (GROUP NOPASS
   ```

4. The following statements must appear in the **VMRMAINT USER RULES** configuration:

```
ACCEPT GSMAPI LINK 193 RR (GROUP NOPASS
```

5. Manager file changes
   - Add the VSMWORK1 machine as a manager.

     For example:

```
MANAGER VSMWORK1 * POOL1  SKELETON VSMWORK1 GENERAL DEVTYPE  VSMWORK1 3390
```

   - Add the user that is passed to SMAPI as a manager. The following example with MAINT:

```
MANAGER MAINT * POOL1
SKELETON MAINT GENERAL
DEVTYPE  MAINT 3390
```

If the security product is not provided by IBM, see the information provided by the independent software vendor.

Appendix I. Configuring IBM Wave for zMON

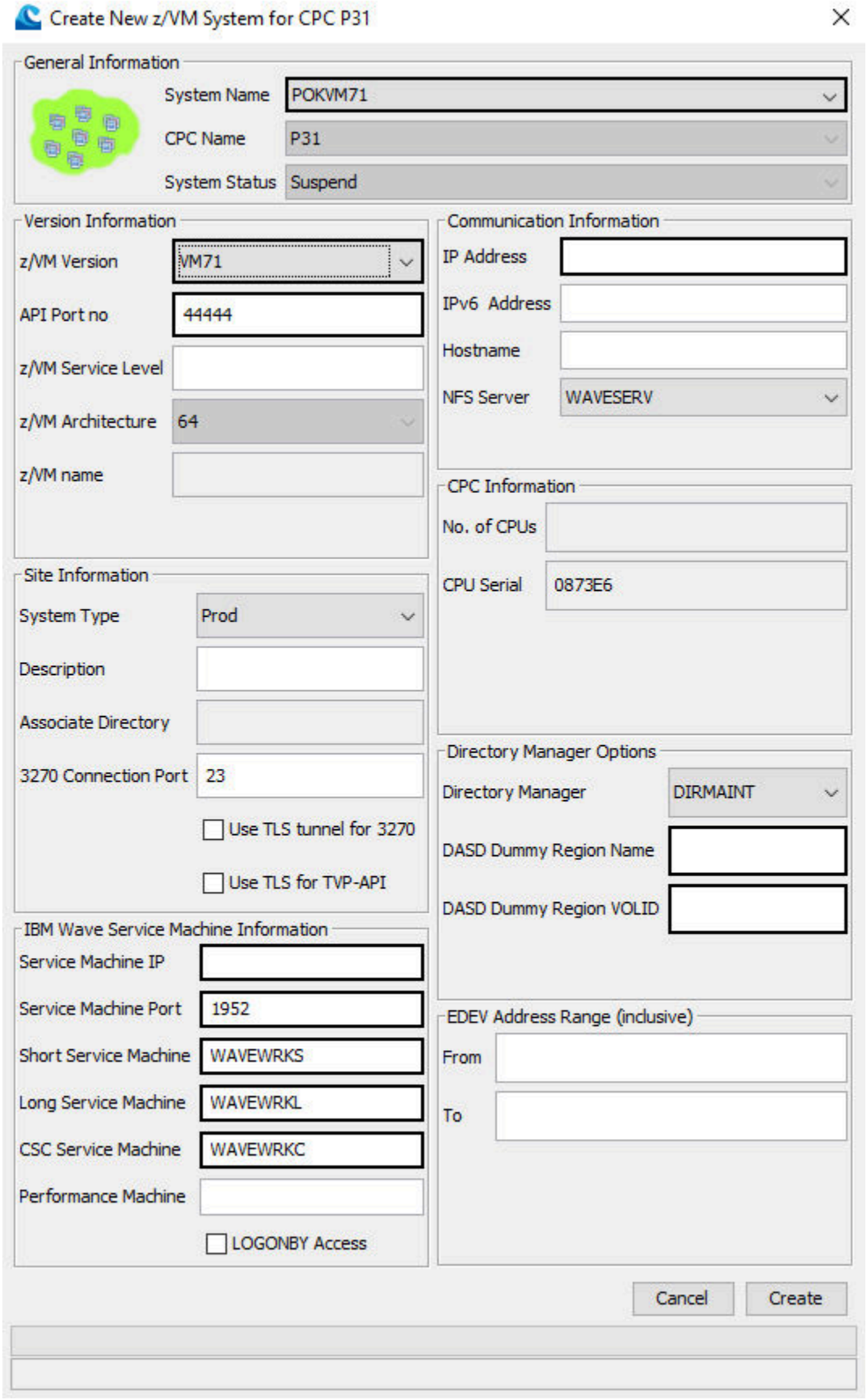
Set the performance service machine

When you add a new z/VM system that runs Velocity Software's zMON performance monitor or change an existing z/VM system to run it, update the IBM Wave Service Machine configuration to specify **ESAMON** as the **Performance Machine**. (Formerly, zMON was known as ESAMON.)

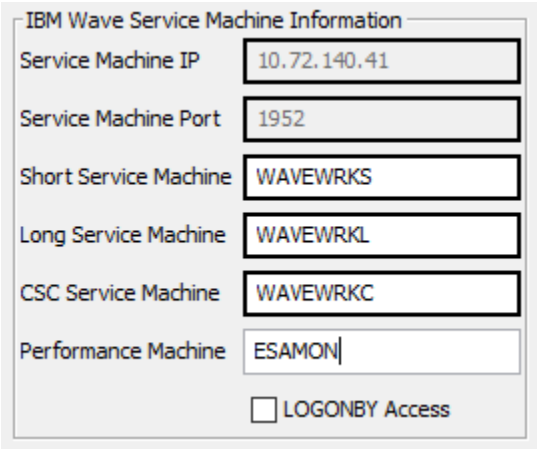
To configure zMON while adding a new z/VM system:

1. Right-click on the CPC's icon in the **Hardware Viewer**.
2. Select **More Actions** from the menu.
3. Select **Add New System** from the menu.

This will display the **Create New z/VM System** window:



In the **IBM Wave Service Machine Information** section at the lower left, enter **ESAMON** in the **Performance Machine** text box:

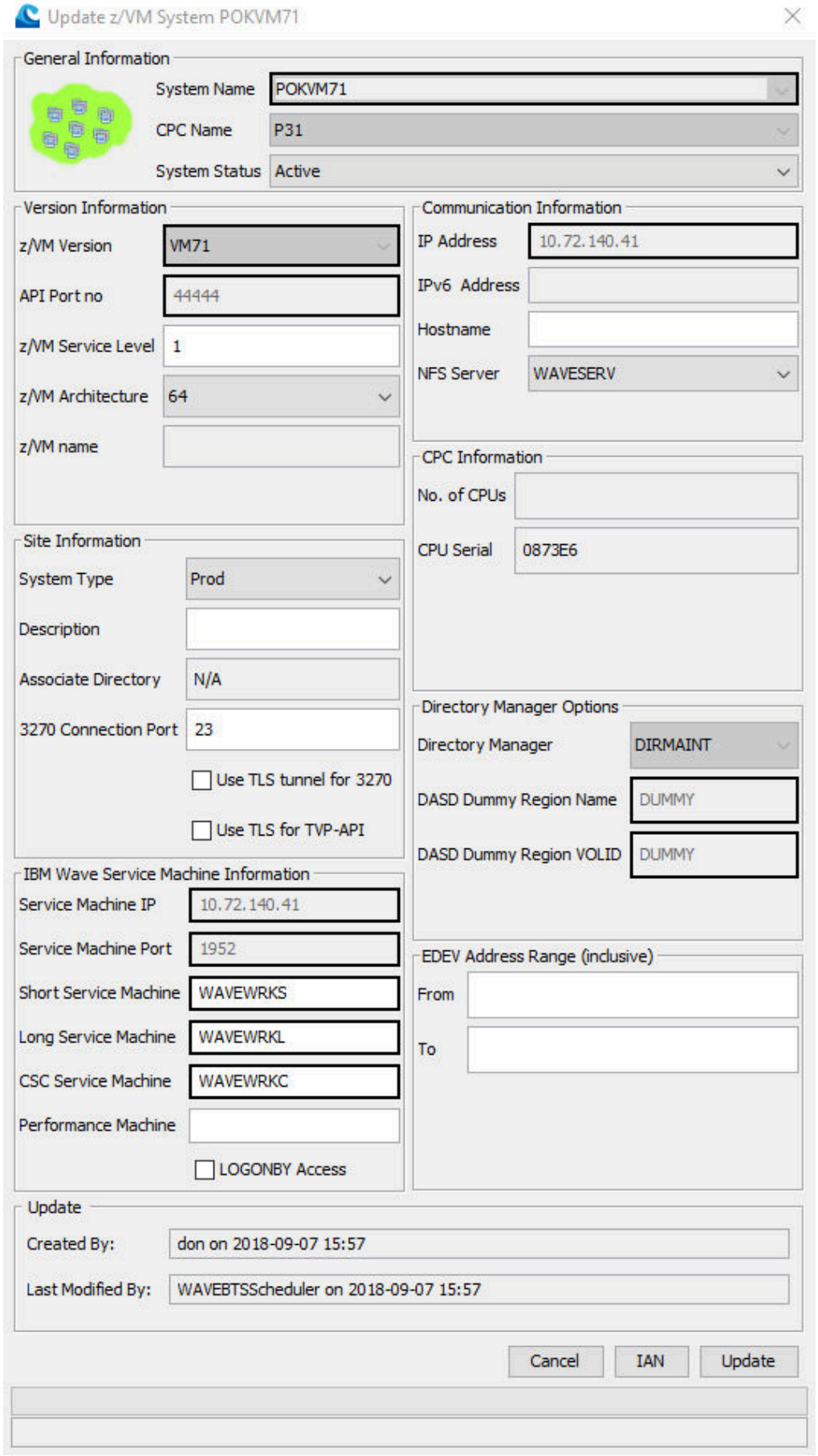


Click on **Create** to complete the configuration change.

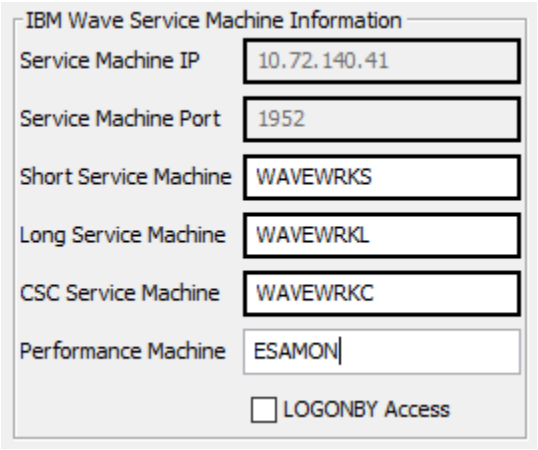
To configure zMON for an existing z/VM system:

1. Right-click on the system's icon in the **Hardware Viewer**.
2. Select **Update Details** from the menu.

This will display the **Update z/VM System window**:


Update z/VM System POKVM71
General Information
System Name
POKVM71
CPC Name
P31
System Status
Active
Version Information
z/VM Version
VM71
API Port no
44444
z/VM Service Level
1
z/VM Architecture
64
z/VM name
Communication Information
IP Address
10.72.140.41
IPv6 Address
Hostname
NFS Server
WAVESERV
CPC Information
No. of CPUs
CPU Serial
0873E6
Site Information
System Type
Prod
Description
Associate Directory
N/A
3270 Connection Port
23
Use TLS tunnel for 3270
Use TLS for TVP-API
Directory Manager Options
Directory Manager
DIRMAINT
DASD Dummy Region Name
DUMMY
DASD Dummy Region VOLID
DUMMY
IBM Wave Service Machine Information
Service Machine IP
10.72.140.41
Service Machine Port
1952
Short Service Machine
WAVEWRKS
Long Service Machine
WAVEWRKL
CSC Service Machine
WAVEWRKC
Performance Machine
LOGONBY Access
EDEV Address Range (inclusive)
From
To
Update
Created By:
don on 2018-09-07 15:57
Last Modified By:
WAVEBTSScheduler on 2018-09-07 15:57
Cancel
IAN
Update

In the **IBM Wave Service Machine Information** section, type **ESAMON** in the **Performance Machine** text box:



Click on **Update** to complete the configuration change.

Establish access to the zMON client

WAVEWRKS uses the ESAMON command to extract performance data and thus needs access to this program and to any associated materials it requires. Update the IBM Wave PROFILE exit, XPRFEXIT EXEC, on the WAVEWRKS 399 minidisk to establish access to the minidisk or SFS directory (typically VMSYSVPS:ZMON.CODE) where these materials reside.

Appendix J. Configuring certificates for managed z/VM systems

When you configure Wave to connect securely to z/VM systems and validate the connections' server certificate, the certificate validation process will fail unless the client side of the connection (the Wave server, and your workstation when using 3270/CLC) trusts the z/VM server certificate's certificate chain.

If the z/VM system's server certificate is signed by a certificate authority (CA) that all clients (including the Wave server and your workstation, as appropriate) in your enterprise trust, no additional configuration is required. In this case, any required certificates have already been installed into the Wave server's JVM and your client workstation by their respective administrators.

In other cases, you must perform manual configuration steps:

- You must add any necessary certificates to the Wave server's JVM, as described in "Adding trusted server certificates to the Wave server" on page 191.
- If you intend to use Wave's 3270 or CLC connection capabilities, you must add any necessary certificates on each workstation running the Wave client application, as described in "Adding trusted server certificates to a Windows workstation" on page 192.

When the z/VM network security administrators export each z/VM system's server certificate, they must choose asn.1, base64 format in gskkyman, and transfer them to their destinations in binary mode.

The number of necessary certificates you need to import will vary based on the size of the certificate chains your z/VM security administrators use, and which certificate authorities already have entries in your enterprise JVM's trusted certificate keystores and in your Windows workstations.

Adding trusted server certificates to the Wave server

You import Secure Socket Layer (SSL) and Transport Layer Security (TLS) certificates into the Wave server's Java keystore of trusted certificates using the **keytool** utility that is supplied with all Java Runtime Environments (JREs):

1. Open a command-line prompt and navigate to the *jre_home_path*/bin directory.
2. Enter the following command:

```
keytool \
-import \
-storepass storepass_password \
-noprompt \
-alias unique_certificate_alias_for_keystore \
-keystore jre_home_path/lib/security/cacerts \
-trustcacerts \
-file path_to_certificate_file
```

Notes:

- The default keystore password for Java is `changeit`.
- The **keytool** utility requires aliases to be unique within a keystore. Wave does not use the alias values, so it imposes no requirements on them.

3. To list the installed certificates, enter one of the following commands:
   a.
   ```
   keytool \
   -keystore jre_home_path/lib/security/cacerts \
   -storepass changeit \
   -list
   ```
   b. This version of the command includes more detailed output:

```
keytool -keystore jre_home_path/lib/security/cacerts -list \
-v | awk '/Alias name:/{print "---";flag=1}/Extensions:/{flag=0}flag'
```

Adding trusted server certificates to a Windows workstation

Use your enterprise's established procedure for adding trusted server certificates to workstations that will run the Wave client application and access managed guests using its 3270 or CLC features.

Notes:

- Many enterprises use policies to prevent you from adding certificates to the JVM running Wave's client application, but if your policy allows adding certificates to the JVM yourself, you can:
  - Open a command prompt with Administrator privileges.
  - Change to the *jre_home_path*/bin directory.
  - Run the commands shown in “Adding trusted server certificates to the Wave server” on page 191.
- Many enterprises that prevent you from adding certificates to the JVM from the command line delegate the certificate store function to Windows. In this case, you can often use Microsoft's Management Console to add the certificates.
  - On Windows 10, typically you should select the local machine store's trusted root certification authorities store, but consult your enterprise's procedures.

Appendix K. Using SSL and TLS certificates for LDAP or Active Directory login

IBM Wave supports LDAP access, which uses Secure Socket Layer and Transport Layer Security (SSL/TLS) cryptographic protocols. When you use SSL/TLS, the certificate for the LDAP system must be imported into a Java keystore on the IBM Wave server. If you need to import additional certificates in order to verify their chain of trust, import the others in the same way.

Use the following steps to create a keystore or to import the certificate into a preexisting keystore:

1. Copy the certificate to the WAVESRV server.
2. Go to `/usr/wave/install`.

   IBM Wave requires that the certificates reside in a keystore in the `waveLdap.jks` directory.
3. To import the SSL/TLS certificate as a trusted certificate, use the **keytool** utility that is supplied with all Java Runtime Environments (JREs):
   a. Open a command-line prompt (on Windows, run the command line as the administrator), and navigate to the `bin` directory.
   b. Enter the following command:

   ```
   "keytool
   -import
   -keystore waveLdap.jks
   -file path_to_certificate_file"
   ```

   Notes:

   - The default keystore password for Java is `changeit`.
   - The **keytool** utility requires aliases to be unique within a keystore. IBM Wave does not use the alias values, so it imposes no requirements on them.
4. To list the installed certificates, enter one of the following commands:
   a.

   ```
   "keytool
   -keystore jre_home_path/lib/security/cacerts
   -storepass changeit
   -list"
   ```

   b. This version of the command includes more detailed output:

   ```
   "keytool -keystore jre_home_path/lib/security/cacerts -list
   -v | awk '/Alias name:/{print "---";flag=1}/Extensions:/{flag=0}flag'"
   ```
5. If the keystore file does not exist, do the following steps:
   a. Follow the prompt and set a password for the keystore and approve adding the certificate.
   b. Log in to IBM Wave with an administrator user and go to **Administrative > Manage Parameters**.

      Note: You need to do this any time you create a new keystore or change the password for an existing keystore.
   c. On the Enterprise Directory tab, enter the new password.
6. If the keystore file exists, do the following steps:
   a. Follow the prompt and enter the current password of the keystore.
   b. Approve adding the certificate to the keystore.

Note: If you do not have the password of the keystore, you can delete the keystore and re-create it with a new password by following the steps for a non-existent keystore file.

Appendix L. Managing Wave's server certificate

When Wave's installation scripts start WebSphere Liberty for the first time:

- Wave's web server component (WAVEWebServer, which uses WebSphere Liberty) creates a Java keystore containing a dynamically-generated, self-signed certificate and the server's automatically-generated public/private key pair. The certificate is used for encryption of the TLS connection between the WebSphere Liberty server and HTTP clients.
- Wave's other component (WAVEBackgroundServices) makes a copy of the web server's certificate and public/private key pair. This component uses the certificate to identify the server when negotiating TLS connections to remote clients like the Wave GUI application. If the copy process fails, this component will dynamically generate its own separate self-signed certificate using a different automatically-generated public/private key pair.

WebSphere Liberty supports two types of keystores that can be used with your own certificates:

1. JKS
2. PKCS12

WebSphere Liberty documentation contains more information about WebSphere Liberty SSL and supported keystores. For more information, see Enabling SSL communication in WebSphere Liberty, SSL configurations, and Keystore configurations for SSL.

Wave keeps the background services component certificate and key pair in a separate internally-generated keystore that you never interact with directly. Its keystore is re-created every time Wave is started. The following topics describe tasks that you might need to perform:

- "Generating a new certificate and signing it" on page 195
- "Viewing the server certificate" on page 198
- "Converting a JKS keystore to PKCS12" on page 198
- "Changing a keystore password" on page 200

Generating a new certificate and signing it

Before you begin

First, you will need to know some background information about certificates.

Things you'll need to change to fit your server

- The Wave server client-resolvable IP address
- If you want Wave users to access it using a domain name, the Wave server client-resolvable host name (discovered during this procedure)
- The command to start and stop Wave.
- Local file names you choose for the certificate signing request (CSR) and certificate, for example.

Java's command-line tool for working with certificates and keystores: keytool

Many of these steps require the use of `keytool` commands. For commands that manipulate a keystore, `keytool` prompts for the password, or you can append `-storepass` *`password_value`* to the command. You can use the `keytool -list` option to see which certificates exist in a store, as well as metadata about each certificate, such as its alias. For more information, see keytool - Key and Certificate Management Tool.

The trailing backslash (\) on long command lines facilitates the copying and pasting of command text from a document to a Linux system. The backslash escapes the carriage return that would otherwise cause the (partial) command to execute immediately (and fail) when it is pasted on the command line.

About this task

The following steps are required to import the certificate into the preexisting JKS keystore.

Procedure

1. On your workstation: discover the host names from which your server is reachable.

   Open a command prompt window and use the `nslookup` and `ping` commands to discover client-resolvable host names matching the Wave server's IP address.

```
nslookup wave_server_ip_address

Server:  server_name
Address:  server_ip_address

Name:    wave_server_host_name
Address:  wave_server_ip_address

ping host_name

Pinging wave_server_host_name [wave_server_ip_address] with 32 bytes of data:
Reply from wave_server_ip_address: bytes=32 time<1ms TTL=59
Reply from wave_server_ip_address: bytes=32 time<1ms TTL=59
Reply from wave_server_ip_address: bytes=32 time=1ms TTL=59
Reply from wave_server_ip_address: bytes=32 time<1ms TTL=59

Ping statistics for wave_server_ip_address:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

2. On the Wave server: generate a server certificate.

   Note: Use commands similar to those shown in the following example. If you are upgrading Wave (rather than performing a new installation) and you want to make a backup of the keystore, make sure you place this backup outside of the `/usr/wave` file tree. Without warning, Wave might remove unrecognized files inside the installation path that it manages.

```
rm /usr/wave/websphere/wlp/usr/servers/defaultServer/resources/security/key.jks

keytool -genkeypair \
-keystore /usr/wave/websphere/wlp/usr/servers/defaultServer/resources/security/key.jks \
-alias default -keyalg RSA -keysize 2048 -dname CN=wave_server_ip_address/host_name \
-storetype jks -validity 365
```

   Note: You can use other supported public key algorithms and key sizes. WebSphere Liberty defaults to RSA-2048 as shown. You *must* use an alias name of `default` (as shown) that must be the only server key in the keystore. When `keytool` prompts for the key password, you *must* use the same value as the keystore password.

3. On the Wave server: generate a CSR for the server's certificate.

```
keytool -certreq \
-keystore /usr/wave/websphere/wlp/usr/servers/defaultServer/resources/security/key.jks \
-alias default -file my.csr -storetype jks
```

4. Obtain a matching certificate that's signed by your enterprise's certificate authority.

   Follow your site's or enterprise's procedure to process the CSR. Your certificate authority (CA) should return a signed certificate — a `.crt` file, for example.

   Note: Your CA might limit the names in the signed certificate. Browsers commonly check the Subject Name and Subject Alternate Name (SAN) fields when validating a server's certificate against the host portion of a URL, but not all CAs support creating SAN fields, for example.

   Use `keytool -list` to compare the signed certificate's attributes against the CSR's attributes so that you are aware of any differences.

The certificate validation checks that browsers run when accessing IBM Wave's launch page, and that Java runs when starting the IBM Wave GUI, are not under IBM Wave's control. These checks depend entirely on the contents of the signed certificate and the code in the browser or JVM that your users are running.

5. On the Wave server: import the new certificate into the keystore.

   Note:

   a. This command will replace the existing certificate. You can make a copy of the `key.jks` file beforehand if you want, or if something goes wrong, you can delete that file and restart WebSphere Liberty to recreate it. The keystore contents are basically disposable. You cannot, however, use a different alias; doing so will cause errors later in the process.

   ```
   keytool -importcert \
   -keystore /usr/wave/websphere/wlp/usr/servers/defaultServer/resources/security/key.jks \
   -alias default -v -file your_file_name -storetype jks
   ```

   b. At least one of the certificate authorities that signed your certificate reply must be trusted by the JVM. If the Wave server's Linux system administrator has not already configured the JVM to trust any of the certificate authorities that signed your certificate reply (typically called *intermediate authorities* or *root authorities*), then before you import your certificate reply, you must configure the JVM to trust those signing authorities. (See The cacerts Certificates File for more information.) Use your site processes to obtain the necessary certificates for the intermediate or root authorities that signed your certificate reply.

   c. In order to successfully import the certificate reply, **keytool** will construct a trust chain. Depending on the format of your site process's certificate reply, you might need to add the **-trustcacerts** parameter to the **keytool -importcert** command invocation above in order for trust chain construction to succeed.

   See Importing a Certificate Reply for more information.

6. On the Wave server: restart Wave to begin using the new certificate.

   Stop and then restart all Wave services by doing one of the following:

   - Re-run Wave's upgrade script (`doUpdate.sh`). If you leave the latest fix pack's files on your Wave Linux server after finishing each upgrade, this might be easier than copying and pasting the lines below.
   - Run the following commands. The restart command will normally generate output only if a problem occurs. If the status command output contains `Active: active (running)` (once for each service), that confirms all services actually started. As long as the restart command generated no messages, your certificate now should be in use.

   ```
   systemctl restart WAVEBackgroundServices WAVEWebServer

   systemctl status WAVEBackgroundServices WAVEWebServer
   ```

7. On your workstation: launch Wave.

   Enter the name of your Wave server on your browser's URL bar.

   The Wave launch page should display in your browser without any "insecure connection" warnings as long as all of the following are true:

   - The Wave server name you used is the Subject Name or one of the Subject Alternate Names in the signed certificate.
   - The signed certificate is in use by Wave.
   - The certificate was signed by a CA that the browser trusts to sign certificates.

   If you get browser warnings displaying the launch page, it confirms that Wave is definitely not using your certificate. In that case, you should review the earlier steps, list the keystore's contents to be sure that you strictly adhered to the documented constraints (see “Viewing the server certificate” on page 198), and use your browser's tools to compare your certificate's attributes against those the

browser checks for in the server's certificate. If your browser displays the launch page without errors, you can continue.

8. On the Wave launch page, left-click on the top button to launch Wave's workstation GUI application.

   This will cause Java to download and execute a Java Web Start file. You might see one or two pop-up windows from Java:

   ```
   Website untrusted
   ```

   or

   ```
   Do you trust content from this publisher?
   ```

   The second pop-up window might always be present. If you check the "Always trust..." box, you'll only see the second pop-up window the first time from any given site; otherwise, you'll see this pop-up window every time. After a few seconds, the Wave application should present a pop-up dialog; see https://www.ibm.com/support/knowledgecenter/SS6JTX/waveug/diag_conn_status.htm for the different pop-ups that could appear and what each one means.

Viewing the server certificate

Procedure

1. The Wave server's Linux administrator(s) can view the contents of Wave's server certificate, as a debugging aid.

   Note: The output of the following command is long enough, and the format opaque enough, that you might want to redirect it to a file and then use `more`, `less`, or an editor to view it. Search for `SubjectAlternativeName` to see the URL host component that a browser will check for a matching value, for example (in `vim` or `vi`): `/subj\c` or `grep -i dns` the output of `keytool -list -v`.

   ```
   keytool -list -v \
   -keystore /usr/wave/websphere/wlp/usr/servers/defaultServer/resources/security/key.jks \
   -storetype jks
   ```

   Use your browser's native tools and controls to figure out its view of things, if the server side looks correct.
2. Copy the certificate to the IBM Wave server.
3. To import the certificate into the keystore, enter:

   ```
   keytool -import -keystore keystore_name -file path_to_certificate_file -alias default \
   -storetype jks
   ```

   a) If the keystore file does not exist, follow the prompt and set a password for the keystore and approve adding the certificate.

   b) If the keystore file exists, follow the prompt and enter the current password of the keystore and approve adding the certificate to the keystore.

Converting a JKS keystore to PKCS12

Before you begin

It is simplest to first follow the procedure used in “Generating a new certificate and signing it” on page 195 to install a server certificate signed by a certificate authority that your enterprise trusts, and then convert the keystore type to PKCS12 when you are sure the new certificate is accepted.

If you choose instead to convert the keystore type before installing your enterprise's server certificate, you must specify `-storetype PKCS12` instead of `-storetype JKS` on the `keytool` commands in "Generating a new certificate and signing it" on page 195.

When you are ready to convert your keystore type to PKCS12, which is considered more secure than the JKS type, you must use a keystore file name (of your choosing). The following example uses a keystore name of `waveLibertyKeystore.p12`.

About this task

The following steps are necessary to import the certificate into the preexisting Public Key Cryptography Standards #12 (PKCS12) keystore.

Procedure

1. On the Wave server: import WebSphere Liberty's current keystore and save it as a new PKCS12 keystore.

   ```
   keytool -importkeystore -srckeystore key.jks -srcstoretype JKS \
   -destkeystore waveLibertyKeystore.p12 -deststoretype PKCS12
   ```

   The `keytool` command will prompt you for the password of the existing JKS keystore and the password of the PKCS12 keystore that you are creating. You *must* use the same passwords throughout this procedure. (After you have verified that your new PKCS12 keystore is working properly, you can change the keystore password later if you want. See "Changing a keystore password" on page 200.)

   The `keytool` command prompts and output should look like this:

   ```
   Enter destination keystore password:
   Re-enter new password:
   Enter source keystore password:
   Entry for alias default successfully imported.
   Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
   ```

2. On the Wave server: edit IBM Wave's Liberty keystore configuration properties file.

   Using a text editor, edit the following file:

   ```
   /usr/wave/install/liberty-bootstrap02.properties
   ```

   Comment out the JKS-related lines by inserting a hash (#) symbol at the start each line, uncomment each corresponding PKCS12-related line by removing its initial hash, and update the keystore file name to match the name you chose.

   IBM Wave's default contents for that file includes lines like this:

   ```
   # Shipped default configuration values
   wave.liberty.keystore.filename=key.jks
   wave.liberty.keystore.type=jks
   # PCKS12 alternative values
   # wave.liberty.keystore.filename=key.p12
   # wave.liberty.keystore.type=PKCS12
   ```

   After making these updates, those lines should look like this:

   ```
   # Shipped default configuration values
   # wave.liberty.keystore.filename=key.jks
   # wave.liberty.keystore.type=jks
   # PCKS12 alternative values
   wave.liberty.keystore.filename= waveLibertyKeystore.p12
   wave.liberty.keystore.type=PKCS12
   ```

3. On the Wave server: restart Liberty to pick up the keystore.

   Stop and then restart the WebSphere Liberty server.

For command syntax, see Starting and stopping the WebSphere Liberty server.

4. On your workstation: launch Wave.

   Enter the name of your Wave server on your browser's URL bar. You should see the same browser prompts or warnings after the conversion that you saw before it.

Results

IBM Wave's WebSphere Liberty uses your PKCS12 keystore file, and users launching the IBM Wave GUI see the same browser prompts that they received when using the JKS keystore. If you need to replace the server certificate later, follow the same procedure used for JKS keystores, remembering to use the correct keystore file name and to specify `-storetype PKCS12` on any `keytool` commands.

Changing a keystore password

Procedure

To change a keystore password, enter:

```
/usr/wave/install/set-keystore-password.sh liberty
```

Appendix M. IBM Wave commands

This topic includes descriptions of IBM Wave commands. For information about the typographic conventions that are used in command syntax, see “Conventions and terminology” on page xix.

WAVEPasswordResetter command

Syntax

```
WAVEPasswordResetter
[-e [ -logfile log_file_name ] ]
[-h]
[-u user_name [ -logfile log_file_name ] ]
```

Description

Use IBM Wave for z/VM's local user password manager, the **WAVEPasswordResetter** command, to manage locally-defined user passwords. Users defined in an enterprise directory such as LDAP must be managed through enterprise-defined means.

In order to run this command, you must be able to log in to the Wave server in a Linux shell with an effective user ID of **root**.

If an unrecognized option is specified, `WAVEPasswordResetter` displays usage information as if the `-h` option was specified.

Options

-e [-logfile *log_file_name*]
Expires all users whose passwords do not comply with IBM Wave's password policy, which prohibits the use of semicolons and requires that passwords be 8-32 characters in length.

If **-logfile** is omitted, it defaults to an invocation-specific name such as `/var/log/WAVE/passwordResetter-2019-07-15_14-22-33.log`.

-h
Displays usage information.

-u *user_name* [-logfile *log_file_name*]
Resets one user's password.

If **-logfile** is omitted, it defaults to an invocation-specific name such as `/var/log/WAVE/passwordResetter-2019-07-15_14-22-33.log`.

Location

`/usr/wave/WAVEBackground`

See also

See also the following topic:

- "The password resetter utility" on page 142

Appendix N. IBM Wave messages

This topic describes the IBM Wave message format and lists the IBM Wave messages for auditable events (HWV prefix).

IBM Wave message format

The following topic describes the format of IBM Wave messages.

IBM Wave issues HWV messages to record audit events. For more information, see “HWVA0001E” on page 203.

Each message consists of the message number, the message text, the source of the message, and the event type.

```
CCCSnnns text
Source
Event Type
```

The HWV message numbers use the following format:

CCCSnnnns
Message number format.

CCC
The three character prefix, **HWV**, that identifies the IBM Wave product.

S
The source identifier, which identifies the IBM Wave subcomponent that produced the message. The subcomponent identifier is one character and is generated by one of the following sources:

A
IBM Wave API server

B
Background Task Scheduler (BTS).

D
Database Restorer.

E
Encryption Key Removal.

G
Graphical user interface (GUI).

P
Password Resetter Utility.

nnnn
The four-digit serial number that identifies the individual message.

s
The message type code is one of the following types:

E
Error - The administrator must act on the message.

I
Information - Informational only. No action is required.

W
Warning - The administrator must review the message and decide whether further action is required.

HWVA0001E	**Missing required field "*JSON field name*".**	**Source:** API Server

Event Type:
API

HWVA0002E **Invalid value was specified for field "*JSON field name*".**

Source:
API Server

Event Type:
API

HWVA0003E **This value cannot be changed by using IBM Wave for z/VM.**

Source:
API Server

Event Type:
API

HWVA0004E **This value cannot be changed via the IBM Wave for z/VM "System" interface.**

Source:
API Server

Event Type:
API

HWVA0005E **The current z/VM System state is "*system state*";
This value can be changed only when the current z/VM system state is "*valid state*".**

Source:
API Server

Event Type:
API

HWVA0006E **This value cannot be changed to *incorrect value for field* via the IBM Wave for z/VM "System" interface.**

Source:
API Server

Event Type:
API

HWVA0007E **Missing required field.**

Source:
API Server

Event Type:
API

HWVA0008E **No HTTP Basic authorization header.**

Source:
API Server

Event Type:
API

HWVA0009E **Malformed HTTP Basic authorization payload.**

Source:
API Server

Event Type:
API

HWVA0010E **CSRF token failure.**

Source:
API Server

Event Type:
API

HWVA0011E **Login failed for *user name*.**

Source:
API Server

Event Type:
API

HWVA0012E **Unauthorized CORS request from *request origin* to *request's HTTP method absolute path of URI*.**

Source:
API Server

Event Type:
API

HWVA0014E **Could not create ENQ for *resource*.**

Source:
API Server

Event Type:
API

HWVA0015E **ETag mismatch between modified and existing z/VM System resource.**

Source:
API Server

Event Type:
API

HWVA0016E **Invalid value was specified.**

Source:
API Server

Event Type:
API

HWVA0017E **A non-empty string value must be specified for this field.**

Source:
API Server

Event Type:
API

HWVA0018E **An integer value must be specified for this field.**

Source:
API Server

Event Type:
API

HWVA0019E **A boolean value must be specified for this field.**

Source:
API Server

Event Type:
API

HWVA0020E **JSON body is missing from POST request.**

Source:
API Server

Event Type:
API

HWVA0021E **JSON body is missing from PUT request.**

Source:
API Server

Event Type:
API

HWVA0022E **Could not load IBM Wave for z/VM Parameters.**

Source:
API Server

Event Type:
API

HWVA0023E **Unable to parse integer parameter for memory minimum.**

Source:
API Server

Event Type:
API

HWVA0024E **Unable to parse integer parameter for memory maximum.**

Source:
API Server

Event Type:
API

HWVA0025E **Unable to parse integer parameter for number of CPUs.**

Source:
API Server

Event Type:
API

HWVA0026E **The provided Automatic Guest Classification (AGC) entry does not match any AGC entry that is defined for this guest.**

Source:
API Server

Event Type:
API

HWVA0027E **Could not get active connection from security context.**

Source:
API Server

Event Type:
API

HWVA0028E **Security context is not an `APISecurityContext`.**

Source:
API Server

Event Type:
API

HWVA0029E **Unable to connect to BTS at host *IP or hostname*:*port*.**

Source:
API Server

Event Type:
API

HWVA0030E **Unable to establish secure connection to BTS at host *IP or hostname***

Source:
API Server

Event Type:
API

HWVA0031I **Login successful.**

Source:
API Server

Event Type:
API

HWVA0032W **The System is going into Single User Mode.**
You are the only user that is allowed to log in.

Source:

API Server

Event Type:
API

HWVA0033W **The System is in a Single User Mode.**
You are the only user that is allowed to log in.

Source:
API Server

Event Type:
API

HWVA0034E **AD Server cannot be contacted.**

Source:
API Server

Event Type:
API

HWVA0035E **IBM Wave for z/VM user is already logged in from *IP or hostname*.**
Log in by using other credentials or force log-off the other instance.

Source:
API Server

Event Type:
API

HWVA0036E **Invalid Domain Name.**

Source:
API Server

Event Type:
API

HWVA0037E **Force log in your IBM Wave for z/VM user failed. Internal error occurred.**
Contact your IBM Wave for z/VM administrator.

Source:
API Server

Event Type:
API

HWVA0038E **The System is going into Single User Mode.**
Contact your IBM Wave for z/VM administrator (*user name*).

Source:
API Server

Event Type:
API

HWVA0039E **Logging in to the system by using an internal IBM Wave for z/VM user name is not allowed.**
Enter another user name.

Source:
API Server

Event Type:
API

HWVA0040E **Invalid Authorization Type.**

Source:
API Server

Event Type:
API

HWVA0041E **User name/Password incorrect.**

Source:
API Server

Event Type:
API

HWVA0042E **Your IBM Wave for z/VM user account is suspended due to too many failed attempts to log in.**
Contact your IBM Wave for z/VM administrator to re-activate your account.

Source:
API Server

Event Type:
API

HWVA0043E **This user is an Active Directory User, you cannot log in with this user name using database login.**

Source:
API Server

Event Type:
API

HWVA0044E **This user is defined as a IBM Wave for z/VM database user, you cannot log in with this user name using LDAP login.**

Source:
API Server

Event Type:
API

HWVA0045E **Your password has expired.**
Contact your IBM Wave for z/VM administrator for assistance.

Source:
API Server

Event Type:
API

HWVA0046E **The System is in a Single User Mode.**
Contact your IBM Wave for z/VM administrator (*user name*).

Source:
API Server

Event Type:
API

HWVA0047E **User is suspended.**
Contact your IBM Wave for z/VM administrator.

Source:
API Server

Event Type:
API

HWVA0048E **Login Failed.**

Source:
API Server

Event Type:
API

HWVA0049E **Could not get SHA-256 message digest instance.**

Source:
API Server

Event Type:
API

HWVA0050E **Invalid project link.**

Source:
API Server

Event Type:
API

HWVA0051E **Invalid Site Defined Group link.**

Source:
API Server

Event Type:
API

HWVA0052E **Invalid device pool link.**

Source:
API Server

Event Type:
API

HWVA0053E **Invalid storage group link.**

Source:
API Server

Event Type:
API

HWVA0054E **Invalid custom attribute link.**

Source:
API Server

Event Type:
API

HWVA0055E **Invalid system link.**

Source:
API Server

Event Type:
API

HWVA0056E **Invalid guest link.**

Source:
API Server

Event Type:
API

HWVA0057E **Invalid volume link.**

Source:
API Server

Event Type:
API

HWVA0058E **Invalid network connection link.**

Source:
API Server

Event Type:
API

HWVA0059E **Invalid virtual switch link.**

Source:
API Server

Event Type:
API

HWVA0060E **Invalid virtual network segment link.**

Source:
API Server

Event Type:
API

HWVA0061E **Invalid real device link.**

Source:
API Server

Event Type:
API

HWVA0062E **Invalid Automatic Guest Classification link.**

Source:
API Server

Event Type:
API

HWVA0063E **The headers If-Modified-Since and Last-Modified are not supported, Use if-match or if-none-match instead.**

Source:
API Server

Event Type:
API

HWVA0064E **Connection to BTS failure. Unsupported BTS Client version *client version*.**

Source:
API Server

Event Type:
API

HWVA0065E **Login Failed. Unsupported return code.**

Source:
API Server

Event Type:
API

HWVA0066E **If-Match header is empty or missing.**

Source:
API Server

Event Type:
API

HWVA0067E **Request must not contain If-Match header.**

Source:
API Server

Event Type:
API

HWVB0010I **Start work unit *work unit name*, token *token ID*, successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Work Unit Internal

HWVB0011I **End work unit *work unit name*, token *token ID*, successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Work Unit Internal

HWVB0051I **SMAPI API *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SMAPI

HWVB0052I **IBM Wave service machine API *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Service Machine API

HWVB0053I **SSH connect by Linux user ID *user* to guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Connection

HWVB0054I **SSH command *command* on guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Commands

HWVB0055I **SSH authorized command *command* on guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Authorized Commands

HWVB0056I **SSH VMCP command *command* on guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SSH VMCP Commands

HWVB0057I **SSH disconnect from guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Connection

HWVB0059I **Command *command name* on Wave Server successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Server Commands

HWVB0101I **Create CPC *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB0102I **Update CPC *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB0103I **Delete CPC *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB0110I **Create z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0111I **Auto-detect z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0112I **Update z/VM system *name* successful, field_name=*field-name*, old value=*old-value*, new value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0113I **Delete z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0114I **Assign directory *name* for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0115I **Shut down started for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0116I **Purge spool for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0117I **Recycle SMAPI for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0118I **Recycle service machine z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0119I **START DASD volume *name* on z/VM system *name* began successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0120I **DRAIN DASD volume *name* on z/VM system *name* began successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0130I **Create directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB0131I **Delete directory *directory-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB0132I **Update directory *directory-name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB0133I **Assign z/VM system *name* to directory *directory-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB0135I **Create network *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB0136I **Update network *type name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB0137I **Delete network *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB0138I **Remove network *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB0139I **Recreate network *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB0140I **Create virtual network segment *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB0141I **Update virtual network segment *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB0142I **Delete virtual network segment *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB0143I **Connect *type name* to virtual network segment *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB0144I **Disconnect *type name* from virtual network segment *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB0150I **Create prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0151I **Update prototype *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0152I **Delete prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0153I **Remove prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0154I **Associate guest *name* prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0155I **Disassociate prototype *name* on z/VM system *name* from guest *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0156I **Remove prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0157I **Duplicate from prototype *name* to prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB0160I **Create storage DASD group *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0161I **Update storage DASD group *name* on directory *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0162I **Delete storage DASD group *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0163I **Remove storage DASD group *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0165I **Create storage DASD region *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0166I **Delete storage DASD region *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0167I **Assign to DASD group *name* storage DASD region/extent from**

DASD volume *name* on directory *name* successful

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0168I **Unassign from DASD group *name* storage DASD region/extent from DASD volume *name* on directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0170I **Remove storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0171I **Vary offline storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0172I **Vary online storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0173I **Add to CP-own list for storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0174I **Attach to *type name* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0175I **Detach from *type name* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0176I **Format storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0177I **Mark as *type* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0178I **Unmark from *type* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0179I **Define DASD region *name* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0180I **Undefine DASD region *name* storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0181I **Assign to DASD group *name* storage for DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0182I **Create storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0183I **Update storage DASD volume *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0184I **Delete storage DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB0190I **Create guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0191I **Directory update for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0192I **Update guest *name* on z/VM system *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0193I **Delete guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0194I **Remove guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0195I **Activate guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0196I **Deactivate guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0197I **Recycle guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0198I **Suspend guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0199I **Resume guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0200I **Connect to VNS *name* VN *type name* guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0201I **Disconnect from VNS *name* VN *type name* guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0202I **Clone of guest definitions from guest *name* on z/VM system *name* to guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0203I **Duplicate z/VM definitions from guest *name* to guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0204I **Init for IBM Wave guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0205I **Refresh Linux data for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0206I **Relocate from z/VM system *name* guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0207I **AGC run guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0208I **Execute REXX *name* guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0209I **Run script *name* guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0210I **Manage storage on guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0211I **Send message to guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0212I **Create account *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0213I **Update account *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0214I **Delete account *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0215I **Add disk space *vdev* size *size type* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0216I **Create *file system type* file system using *name* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0217I **Create logical volume *name* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0218I **Add new CKD minidisk *vdev* size *size* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0219I **Add new EDEV minidisk *vdev* size *size* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0220I **Create volume group *name* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0221I **Resize file system *name* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0222I **Set boot device *vdev name* for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0223I **Create *name* partition for guest *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0230I **Clone using DDR *vdev* for guest *name* from *vdev* for *guest* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0231I **Clone using directory manager minidisk *vdev* for guest *name* from *vdev* for *guest* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0232I **Cross system clone *vdev* for guest *name* on z/VM system *name* from *vdev* for *guest* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0233I **Dedicate device *addr* for guest *name* as *vdev* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0240I **Start Linux installation for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0241I **Write Linux installation parameters for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0242I **Reset Linux installation for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0243I **Linux installation on guest *name* complete. Check guest for completion information and status.**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0245I **Create IBM Wave service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0246I **Deactivate IBM Wave service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0247I **Activate IBM Wave service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0248I **Populate IBM Wave service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0249I **Verify IBM Wave service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0250I **Build AUTOLOG service machine *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0251I **Activate schedule for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0252I **Dummy region *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0270I **Regenerate IBM Wave encryption key successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0271I **Log in from *PC name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0272I **Log out successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0273I **Terminate successful**

Source:
Background Task Scheduler (BTS)

Event Type:

Security

HWVB0274I **Forced log out for IBM Wave user *user name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0276I **Change TVP-API user password successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0277I **Set service machine minidisk passwords in database successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0278I **Verify request *request* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0279I **Verify TVP-API for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0280I **Suspend TVP-API for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0281I **Regenerate IBM Wave user SSH pair key successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0282I **Change Password for IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB0292I **Add IBM Wave work unit worker successful; now there are *number* workers**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0293I **Remove IBM Wave work unit worker successful; now there are *number* workers**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0294I **Delete IBM Wave work unit *work unit name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0295I **Reset BTS statistics until next error occurred for *BTS request type BTS request parameters* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0296I **Rotate IBM Wave system COR new system COR *new COR* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0297I **Delete IBM Wave system COR *COR name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0298I **Clean IBM Wave system COR successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0299I **Deactivate IBM Wave schedule entry *name* for *parameter* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0300I **Activate IBM Wave schedule entry *name* for *parameter* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0301I **Change IBM Wave schedule entry *name* for *parameter* successful, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0302I **Deactivate all IBM Wave schedule entries for directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0303I **Activate all IBM Wave schedule entries for directory *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0304I **Deactivate all IBM Wave scheduled entries for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0305I **Activate all IBM Wave schedules for z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0306I **Back up IBM Wave knowledge base to file name *file-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB0308I **Change ignore BTS statistics until next error occurred for *BTS request type BTS request parameters* successful, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler

HWVB0309I **Change permanently ignore BTS statistics for *BTS request type BTS request parameters* successful, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler

HWVB0310I **Suspend IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0311I **Suspend IBM Wave user *name* successful; invalid password entered *number* times**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0312I **Suspend IBM Wave user *name* successful; ID not used for *days* days**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0313I **Create IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0314I **Update IBM Wave user *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0315I **Add scope *scope* permission *permission* for IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0316I **Delete scope *scope* permission *permission* for IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0317I **Update scope *scope* permission *permission* for IBM Wave user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0318I **Create IBM Wave profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0319I **Update IBM Wave profile *file name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0320I **Delete IBM Wave profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0321I **Connect IBM Wave profile *file name* to LDAP group *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0322I **Disconnect IBM Wave profile *file name* from LDAP group *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0323I **Add scope *scope* permission *permission* to IBM Wave profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0324I **Delete scope *scope* permission *permission* from IBM Wave profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0325I **Update scope *scope* permission *permission* for IBM Wave profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0326I **Clone from IBM Wave user *name* to user *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0327I **Clone from IBM Wave profile *file name* to profile *file name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0328I **Delete IBM Wave user *user name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB0330I **Create IBM Wave project *project name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0331I **Update IBM Wave project *project name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0332I **Delete IBM Wave project *project name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0333I **Create IBM Wave site define group *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0334I **Update IBM Wave site define group *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0335I **Delete IBM Wave site define group *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0336I **Create AGC entry *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0337I **Update AGC entry *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0338I **Delete AGC entry *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0339I **Change status of AGC entry *name* from *old-value* to *new-value* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0340I **Create custom attribute *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0341I **Update custom attribute *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0342I **Delete custom attribute *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB0350I **Create Linux repository *repository-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB0351I **Update Linux repository *repository-name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB0352I **Delete Linux repository *repository-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB0353I **Discover Linux repository *repository-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB0354I **Verify Linux repository *repository-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB0360I **Create device pool *name* type *type* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0361I **Update device pool *name* type *type* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0362I **Remove device pool *name* type *type* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0363I **Associate device pool *name* to z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0364I **Disassociate device pool *name* from z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0365I **Transfer real device *name* from device pool *old-pool* to device pool *new-pool* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB0366I **Create managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0367I **Update managed entity *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0368I **Remove managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0369I **Connect VNS *name* to managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0370I **Connect device pool *name* to managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0371I **Disconnect VNS *name* from managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0372I **Disconnect device pool *name* from manage entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0373I **Connect WWPN *WWPN* to managed entity *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB0380I **Ignore attention required *attn* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB0381I **Unignore attention required *attn* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB0382I **Change attention required *attn* severity from *old-severity* to *new-severity* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB0383I **Reset attention required *attn* severity from *old-severity* to default *default-severity* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB0384I **Reset attention required *attn* ignore from *old-severity* to default *default-severity* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB0390I **Create IBM Wave report template *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB0391I **Update IBM Wave report template *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB0392I **Delete IBM Wave report template *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB0393I **Generate IBM Wave report template *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB0395I **Create script *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB0396I **Update script *name* successful, field_name=*field-name*, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB0397I **Delete script *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB0398I **IBM Wave script NFS synchronized successful**

Source:
Background Task Scheduler (BTS)

Source:

Event Type:
Scripts

HWVB0400I **Change IBM Wave parameter *name* successful, old_value=*old-value*, new value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB0401I **Change IBM Wave security parameter *name* successful, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB0402I **Change IBM Wave audit parameter *name* successful, old_value=*old-value*, new_value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB0403I **Clean work unit *work unit name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB0404I **Delete work unit *work unit name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB0405I **Delete IBM Wave COR entry *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB0406I **Clean BTS work units successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB0407I **Truncate of audit messages older than *number* days to file *file-name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Audit

HWVB0410I **Broadcast message successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Actions

HWVB0415I **Retrieve details for guest *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0416I **Retrieve signal activation done for guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0417I **Display guests by filter *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB0418I **Display z/VM systems by filter *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB0419I **Retrieve workunit status for workunit *number* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB1000I **Discovered update of z/VM system *name* successful, field_name=*field-name*, old value=*old-value*, new value=*new-value***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB1002I **Discovered guest *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1003I **Discovered update of guest *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1004I **Discovered delete of guest *name* on z/VM system *name* successful; marked inconsistent**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1006I **Discovered prototype *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB1007I **Discovered update of prototype *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB1008I **Discovered delete of prototype *name* on z/VM system *name* successful, marked inconsistent**

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB1010I **Discovered network *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB1011I **Discovered update network *type name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB1012I **Discovered delete network *type name* on z/VM system *name* successful; marked inconsistent**

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB1014I **Discovered DASD group *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1015I **Discovered update of DASD group *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1016I **Discovered delete of DASD group *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1018I **Discovered DASD region *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1019I **Discovered update of DASD region *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1020I **Discovered delete of DASD region *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1022I **Discovered DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1023I **Discovered update of DASD volume *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1024I **Discovered delete of DASD volume *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB1026I **Discovered real device *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB1027I **Discovered update of real device *type name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB1028I **Discovered delete of real device *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB1030I **Discovered profile *file name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1031I **Discovered update of profile *file name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:

Guest

HWVB1032I **Discovered delete of profile *file name* on z/VM system *name* successful; marked inconsistent**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1035I **Discovered account *name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1036I **Discovered update of account *name* on z/VM system *name* successful, field_name=*field-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1037I **Discovered delete of account *name* on z/VM system *name* successful; marked inconsistent**

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB1040I **Discovered dedicate of real device *type name* to *type name* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB1041I **Discovered dedicate remove of real device *type name* from *typename* on z/VM system *name* successful**

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB5011E **Start work unit *work unit name*, token *token ID*, aborted**

Source:
Background Task Scheduler (BTS)

Event Type:
Workunit-Internal

HWVB5051E **SMAPI API *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SMAPI

HWVB5052E **IBM Wave service machine API *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Service Machine API

HWVB5053E **SSH connect by Linux user ID *user* to guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Connection

HWVB5054E **SSH command *command* on guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Commands

HWVB5055E **SSH authorized command *command* on guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Authorized Commands

HWVB5056E **SSH VMCP command *command* on guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SSH VMCP Commands

HWVB5057E **SSH disconnect from guest *name* on z/VM system *name* failed, reason=*reason***

Source:

Background Task Scheduler (BTS)

Event Type:
SSH Connection

HWVB5058E **SSH terminate from guest *name* on z/VM system *name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
SSH Connection

HWVB5059E **Command *command name* on Wave Server failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Server Commands

HWVB5101E **Create CPC *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB5102E **Update CPC *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB5103E **Delete CPC *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
CPC

HWVB5110E **Create z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5111E **Auto-detect z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5112E **Update z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5113E **Delete z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5114E **Assign directory *name* for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5115E **Shut down for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5116E **Purge spool for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5117E **Recycle SMAPI for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5118E **Recycle service machine for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5119E **START DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5120E **DRAIN DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5130E **Create directory *directory-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB5131E **Delete directory *directory-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB5132E **Update directory *directory-name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB5133E **Assign z/VM system *name* to directory *directory-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Directory

HWVB5135E **Create network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB5136E **Update network *type name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB5137E **Delete network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB5138E **Remove network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB5139E **Recreate network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB5140E **Create virtual network segment *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB5141E **Update virtual network segment *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB5142E **Delete virtual network segment *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:

Virtual Network Segment

HWVB5143E **Connect *type name* to virtual network segment *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB5144E **Disconnect *type name* from virtual network segment *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Virtual Network Segment

HWVB5150E **Create prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5151E **Update prototype *name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5152E **Delete prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5153E **Remove prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5154E **Associate guest *name* prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5155E **Disassociate prototype *name* on z/VM system *name* from guest *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5156E **Remove prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5157E **Duplicate from prototype *name* to prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB5160E **Create storage DASD group *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5161E **Update storage DASD group *name* on directory *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5162E **Delete storage DASD group *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5163E **Remove storage DASD group *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5165E **Create storage DASD region *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5166E **Delete storage DASD region *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5167E **Assign to DASD group *name* storage DASD region/extent from DASD volume *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5168E **Unassign from DASD group *name* storage DASD region/extent from DASD volume *name* on directory *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5170E **Remove storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5171E **Vary offline storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5172E **Vary online storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5173E **Add to CP-own list for storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5174E **Attach to *type name* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5175E **Detach from *typename* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5176E **Format storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5177E **Mark as *type* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5178E **Unmark from *type* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5179E **Define DASD region *name* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5180E **Undefine DASD region *name* storage DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5181E **Assign to DASD group *name* storage for DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5182E **Create storage DASD volume *name EDEV* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5183E **Update storage DASD volume *name EDEV* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5184E **Delete storage DASD volume *name EDEV* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB5190E **Create guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5191E **Directory update for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5192E **Update guest *name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5193E **Delete guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5194E **Remove guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5195E **Activate guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5196E **Deactivate guest *name* on z/VM system *name***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5197E **Recycle guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5198E **Suspend guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5199E **Resume guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5200E **Connect to VNS *name* VN *type name* guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5201E **Disconnect from VNS *name* VN *type name* guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5202E **Clone of guest definitions from guest *name* on z/VM system *source guest system* to guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5203E **Duplicate z/VM definitions from guest *name* to guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5204E **Init for IBM Wave guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5205E **Refresh Linux data for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5206E **Relocate from z/VM system *name* guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5207E **AGC run guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5208E **Execute REXX *name* guest *name* on z/VM system *name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:

Guest

HWVB5209E **Run script *name* guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5210E **Manage storage on guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5211E **Send message to guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5212E **Create account *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5213E **Update account *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5214E **Delete account *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5215E **Add disk space *vdev* size *size type* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5216E **Create *file system name* file system *name* for guest using *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5217E **Create logical volume *name* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5218E **Add new CKD minidisk *vdev* size *size* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5219E **Add new EDEV minidisk *vdev* size *size* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5220E **Create volume group *name* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5221E **Resize file system *name* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5222E **Set boot device *vdev name* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5223E **Create *name* partition for guest *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5230E **Clone using DDR *vdev* for guest *name* from *vdev* for *guest* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5232E **Cross system clone *name* for guest *name* on z/VM system *name* from *name* for *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5231E **Clone using directory manager minidisk *vdev* for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Clone

Event Type:
Background Task Scheduler (BTS)

HWVB5233E **Dedicate device *addr* for guest *name* as *vdev* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5240E **Start Linux installation for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5241E **Write Linux installation parameters for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5242E **Reset Linux installation for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5243E **Linux installation on guest *guest* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5245E **Create IBM Wave service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5246E **Deactivate IBM Wave service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5247E **Activate IBM Wave service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5248E **Populate IBM Wave service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5249E **Verify IBM Wave service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5250E **Build AUTOLOG service machine *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5251E **Activate schedule for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5252E **Dummy region *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5270E **Regenerate IBM Wave encryption key failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5271E **Log in from *PC name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5272E **Log out failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5273E **Terminate failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5274E **Forced log out for IBM Wave user *user name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5275E **Invalid password from *PC-name***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5276E **Change TVP-API user password failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5277E **Set service machine minidisk passwords in database failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5278E **Verify request *request* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5279E **Verify TVP-API for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5280E **Suspend TVP-API for z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5281E **Regenerate IBM Wave user SSH pair key failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5282E **Change Password for IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Security

HWVB5292E **Add IBM Wave work unit worker failed; now there are *number* workers**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5293E **Remove IBM Wave work unit worker failed; now there are *number* workers**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5294E **Delete IBM Wave work unit *work unit name* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5295E **Reset BTS statistics until next error occurred for *BTS request type BTS request parameters* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5296E **Rotate IBM Wave system COR new system COR *COR name* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5297E **Delete IBM Wave system COR *new COR* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5298E **Clean IBM Wave system COR failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5299E **Deactivate IBM Wave schedule entry *name* for *parameter* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5300E **Activate IBM Wave schedule entry *name* for *parameter* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5301E **Change IBM Wave schedule entry *name* for *parameter* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5302E **Deactivate all IBM Wave schedule entries for directory *name* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5303E **Activate all IBM Wave schedule entries for directory *name* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5304E **Deactivate all IBM Wave scheduled entries for z/VM system *name* failed**

Source:

Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5305E **Activate all IBM Wave schedules for z/VM system *name* failed**

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5306E **Back up IBM Wave knowledge base to file name *file-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5308E **Change ignore BTS statistics until next error occurred for *BTS request type BTS request parameters* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler

HWVB5309E **Change permanently ignore BTS statistics for *BTS request type BTS request parameters* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler

HWVB5310E **Suspend IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5311E **Suspend IBM Wave user *name* failed; invalid password entered *number* times, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5312E **Suspend IBM Wave user *name* failed; ID not used for *days* days, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5313E **Create IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5314E **Update IBM Wave user *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5315E **Add scope *scope* permission *permission* for IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5316E **Delete scope *scope* permission *permission* for IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5317E **Update scope *scope* permission *permission* for IBM Wave user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5318E **Create IBM Wave profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5319E **Update IBM Wave profile *file name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Background Task Scheduler (BTS)

HWVB5320E **Delete IBM Wave profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5321E **Connect IBM Wave profile *file name* to LDAP group *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5322E **Disconnect IBM Wave profile *file name* from LDAP group *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5323E **Add scope *scope* permission *permission* to IBM Wave profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5324E **Delete scope *scope* permission *permission* from IBM Wave profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5325E **Update scope *scope* permission *permission* for IBM Wave profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5326E **Clone from IBM Wave user *name* to user *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5327E **Clone from IBM Wave profile *file name* to profile *file name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5328E **Delete IBM Wave user *user name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Management

HWVB5330E **Create IBM Wave project *project name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5331E **Update IBM Wave project *project name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5332E **Delete IBM Wave project *project name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5333E **Create IBM Wave site define group *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5334E **Update IBM Wave site define group *name* failed, field_name=*field-name*, reason=*reason***

Source:

Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5335E **Delete IBM Wave site define group *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5336E **Create AGC entry *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5337E **Update AGC entry *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5338E **Delete AGC entry *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5339E **Change status of AGC entry *name* from *old-status* to *new-status* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5340E **Create custom attribute *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5341E **Update custom attribute *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5342E **Delete custom attribute *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest Management

HWVB5350E **Create Linux repository *repository-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB5351E **Update Linux repository *repository-name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB5352E **Delete Linux repository *repository-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB5353E **Discover Linux repository *repository-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB5354E **Verify Linux repository *repository-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Linux Repository

HWVB5360E **Create device pool *name* type *type* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5361E **Update device pool *name* type *type* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5362E **Remove device pool *name* type *type* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5363E **Associate device pool *name* to z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5364E **Disassociate device pool *name* from z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5365E **Transfer real device *name* from device pool *old-pool* to device pool *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Device Pools

HWVB5366E **Create managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5367E **Update managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5368E **Remove managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5369E **Connect VNS *name* to managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5370E **Connect device pool *name* to managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5371E **Disconnect VNS *name* from managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5372E **Disconnect device pool *name* from manage entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5373E **Connect WWPN *WWPN* to managed entity *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Managed Entities

HWVB5380E **Ignore attention required *attn* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB5381E **Remove ignore attention required *attn* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB5382E **Change attention required *attn* severity from *old-severity* to *new-severity* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB5383E **Reset attention required *attn* severity from *old-severity* to default *default-severity* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB5384E **Reset attention required *attn* ignore from *old-ignore* to default *default-ignore* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Attention Required

HWVB5390E **Create IBM Wave report template *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB5391E **Update IBM Wave report template *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB5392E **Delete IBM Wave report template *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB5393E **Generate IBM Wave report template *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Reports

HWVB5395E **Create script *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB5396E **Update script *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB5397E **Delete script *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB5398E **IBM Wave script NFS synchronized failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Scripts

HWVB5400E **Change IBM Wave parameter *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB5401E **Change IBM Wave security parameter *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB5402E **Change IBM Wave audit parameter *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB5403E **Clean work unit *work unit name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Parameters

HWVB5404E **Delete work unit *work unit name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB5405E **Delete IBM Wave COR entry *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB5406E **Clean BTS work units failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB5407E **Truncate of audit messages older than *number* days to file *file-name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Audit

HWVB5410E **Broadcast message failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave User Actions

HWVB5415E **Retrieve details for guest *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5416E **Retrieve signal activation done for guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5417E **Display guests by filter *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB5418E **Display z/VM system by filter *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB5419E **Retrieve workunit status for workunit *number* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Wave Workunit

HWVB6000E **Discovered update of z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
System

HWVB6002E **Discovered guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6003E **Discovered update of guest *name* on z/VM system *name* failed, field_name=*field-name* reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6004E **Discovered delete of guest *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6006E **Discovered prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB6007E **Discovered update of prototype *name* on z/VM system *name* failed, field_name=*field-name* reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB6008E **Discovered delete of prototype *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Prototype

HWVB6010E **Discovered network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB6011E **Discovered update network *type name* on z/VM system *name* failed, field_name=*field-name* reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB6012E **Discovered delete network *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Network

HWVB6014E **Discovered DASD group *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6015E **Discovered update of DASD group *name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6016E **Discovered delete of DASD group *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6018E **Discovered DASD region *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6019E **Discovered update of DASD region *name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6020E **Discovered delete of DASD region *name* on z/VM system *name* failed, reason=*reasony***

Source:

Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6022E **Discovered DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6023E **Discovered update of DASD volume *name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6024E **Discovered delete of DASD volume *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Storage

HWVB6026E **Discovered real device *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB6027E **Discovered update of real device *type name* on z/VM system *name* failed, field_name=*field-name*, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB6028E **Discovered delete of real device *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB6030E **Discovered profile *file name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6031E **Discovered update of profile *file name* on z/VM system *name* failed, field_name=*field-name* reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6032E **Discovered delete of profile *file name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6035E **Discovered account *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6036E **Discovered update of account *name* on z/VM system *name* failed, field_name=*field-name* reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6037E **Discovered delete of account *name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Guest

HWVB6040E **Discovered dedicate of real device *type name* to *type name* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVB6041E **Discovered dedicate remove of real device *type name* from *typename* on z/VM system *name* failed, reason=*reason***

Source:
Background Task Scheduler (BTS)

Event Type:
Real Device

HWVC0001I **Request was verified.**

Source:
API Server

Event Type:
API

HWVC0002E **Failed to verify request. Check parameters.**

Source:
API Server

Event Type:
API

HWVC0003E **Failed to verify request. User is unauthorized.**

Source:
API Server

Event Type:
API

HWVC0004E **The specified resource is ENQd by another user.**

Source:
API Server

Event Type:
API

HWVC0005E **The specified resource is locked.**

Source:
API Server

Event Type:
API

HWVC0006E **The specified resource exists and cannot be created.**

Source:
API Server

Event Type:
API

HWVC0007E **The specified parameters are incorrect.**

Source:
API Server

Event Type:
API

HWVC0008E **The specified resource does not exist.**

Source:
API Server

Event Type:
API

HWVC0009E **Internal error occurred during processing of the request.**

Source:
API Server

Event Type:
API

HWVC0010E **Request could not be processed by remote BTS.**

Source:
API Server

Event Type:
API

HWVC0011E **Verifier not found for request.**

Source:
API Server

Event Type:
API

HWVC0012E **The specified resource is inconsistent.**

Source:
API Server

Event Type:
API

HWVC0013E **The specified resource is not eligible for this action while in its current state.**

Source:
API Server

Event Type:
API

HWVC0014E **BTS request failed with return code: *return code*.**

Source:
API Server

Event Type:
API

HWVC0015E **Failed to verify request. Scope or permission error.**

Source:
API Server

Event Type:
API

HWVD0001I **Restore database from file *file name* successful**

Source:
Database Restorer

Event Type:
Security

HWVD5001E **Restore database from file *file name* failed, reason=*reason***

Source:
Database Restorer

Event Type:
Security

HWVE0001I **Wave Encryption Key successful**

Source:
Encryption Key Removal

Event Type:
Security

HWVE5001E **Wave Encryption Key failed, reason=*reason***

Source:
Encryption Key Removal

Event Type:
Security

HWVG0001I **Back up of audit messages to file *file name* successful**

Source:
GUI

Event Type:
GUI actions

HWVG0002I **Truncate of audit messages older than *days* days to file *file name* successful**

Source:
GUI

Event Type:
GUI

HWVG0003I **Import file *file name* successful**

Source:
GUI

Event Type:
GUI

HWVG0004I **Generate IBM Wave report *report-name* type *report-type* successful**

Source:
GUI

Event Type:
GUI

HWVG0010I **Enable single user mode for IBM Wave user *name* successful**

Source:
GUI

Event Type:
GUI

HWVG0011I **Disable single user mode for IBM Wave user *name* successful**

Source:
GUI

Event Type:
GUI actions

HWVG0012I **Force DEQ for WRS element *element name* successful**

Source:
GUI

Event Type:
GUI

HWVG0015I **Change of IBM Wave database password successful**

Source:
GUI

Event Type:
Security

HWVG5001E **Back up of audit message to file *file name* failed, reason=*reason***

Source:
No group

Event Type:
GUI

HWVG5002E **Truncate log of audit messages older than *days* days to file *file name* failed, reason=*reason***

Source:
No group

Event Type:
GUI actions

HWVG5004E **Generate IBM Wave report *report-name* type *report-type* failed, reason=*reason***

Source:
GUI

Event Type:
GUI

HWVG5010E **Enable single user mode for IBM Wave user *name* failed, reason=*reason***

Source:
GUI

Event Type:
GUI

HWVG5011E **Disable single user mode for IBM Wave user *name* failed, reason=*reason***

Source:
GUI

Event Type:
GUI

HWVG5012E **Force DEQ for WRS element *element name* failed, reason=*reason***

Source:
GUI

Event Type:
GUI

HWVG5015E **Change of IBM Wave database password failed, reason=*reason***

Source:
GUI

Event Type:
Security

HWVP0001I **Reset IBM Wave user *user_name* successful**

Source:
Password Resetter Utility

Event Type:
Security

HWVP0100I **Save IBM Wave *unique_name* keystore password successful**

Source:
Password Resetter Utility

Event Type:
Security

HWVP5001E **Reset IBM Wave user *user_name* failed, reason=*reason***

Source:
Password Resetter Utility

Event Type:
Security

HWVP5100E **Save IBM Wave *unique_name* keystore password failed**

Source:
Password Resetter Utility

Event Type:
Security

Appendix O. IBM Wave user exits

IBM Wave provides user exits for your installation to be able to customize and receive control before or after an IBM Wave action is run.

Linux managed z/VM guest exits
Linux executable exits that are stored on managed z/VM guests.

Linux Wave server exits
Linux executable exits that are stored on the Wave server.

REXX exits
The REXX exits, which are written in REXX, must be stored in the 399 minidisk on the WAVEWRKS service machine.

WAVECloneConfigExit - Cloned server first boot exit

Use the `WAVECloneConfigExit.sh` exit to customize Linux z/VM Guests during the first boot after the guests are cloned from another z/VM Guest or prototype.

Point of processing
This shell script is run using bash immediately after IBM Wave changes the IP addresses, host name, and other aspects of the cloned z/VM guest (at the first boot).

Exit code location
A shell script that is named `WAVECloneConfigExit.sh` is stored in the `/usr/wave/exit` directory on the source z/VM Guest.

Parameters sent to this exit
None.

Return Codes
IBM Wave does not check the return code from this exit.

WAVENetConfigExit - Connect or disconnect processing

Use the `WAVENetConfigExit.sh` exit to customize Linux z/VM Guests after a successful connect or disconnect action is issued against the guests.

Point of processing
None.

Exit code location
A shell script named `WAVENetConfigExit.sh` must be in the `/usr/wave/exit` directory on the source z/VM guest.

Parameters sent to the exit
None.

Return Codes
IBM Wave does not check the return code from this exit.

WaveConnectableGuestsExit - Connectable guests exit

Description
The `WaveConnectableGuestsExit` can be used to test which guests are connectable.

Point of processing
When the `WaveConnectableGuestsExit` is in the `/usr/wave/exit/` directory, IBM Wave automatically uses the exit to test if guests are connectable.

When the exit is not present, IBM Wave internally tests if guests are connectable.

Exit code location
An executable file that is named `WaveConnectableGuestsExit` must be in the `/usr/wave/exit/` directory on the Wave server. The permissions for the exit file must be set to 700, and the owner must be waveuser.

Exit Input
The `WaveConnectableGuestsExit` receives the following input parameters:

Port
SSH port number.

IP addresses
IP addresses that are passed as individual arguments.

Exit Output
The `WaveConnectableGuestsExit` must output a list of the connectable IP addresses that are separated by an end of line character ("\n").

Timeout Value
To calculate the timeout value, multiply the number of IP addresses by the "SSH timeout value" that is defined in the **Administrative** > **Manage Parameters** > **Functionality** tab. The minimum timeout value is ten seconds. The `WaveConnectableGuestsExit` must complete its run within the timeout value. If not, the exit's output is ignored.

Return Codes
IBM Wave does not check the return code from this exit.

Sample Exit
A sample exit, `WaveConnectableGuestsExit.samp`, is in the `/usr/wave/exit/` directory. `WaveConnectableGuestsExit.samp` uses the **nmap** command. If you use the sample, ensure that **nmap** is installed on the IBM Wave server.

Example Input and Output
The following example shows the sample input and output for the `WaveConnectableGuestsExit` when its run to check the connectivity of the following three guests on port 22:

`198.51.100.2` (connectable).
`203.0.113.0` (not connectable).
`198.51.100.15` (connectable).

The exit is executed as:

```
/usr/wave/exit/WaveConnectableGuestsExit 22 198.51.100.2 203.0.113.0 198.51.100.15
```

The output is as follows:

```
198.51.100.2
198.51.100.15
```

XPRFEXIT - PROFILE EXEC exit for service machines

Use the XPRFEXIT exit to add processing to the PROFILE EXEC that is shared by the IBM Wave service machines.

Point of processing
The XPRFEXIT exit is called by the shared PROFILE EXEC of IBM Wave service machines after the service machine minidisks are accessed and other configuration is complete. After complete execution of the XPRFEXIT routine, the PROFILE EXEC proceeds to initialize the IBM Wave service machine server code based on the return value from XPRFEXIT.

By default, IBM Wave does not ship a sample of the XPRFEXIT file.

1. Copy the sample REXX EXEC file Figure 82 on page 252.

2. Create a new file named "XPRFEXIT EXEC" on the 399 disk of the WAVEWRKS service machine, and then modify it as necessary.

Important: Any user script can be supplied provided it meets the requirements that are outlined in "**Exit code location**", "**Parameters sent to the exit**", and "**Return Codes**" sections. Any deviation from the requirements is not supported.

Exit code location

The XPRFEXIT exit must be copied and stored to the 399 minidisk of the WAVEWRKS service machine. The file name must be "XPRFEXIT EXEC".

Parameters sent to the exit

None.

Return Codes

The XPRFEXIT exit must return a valid return code.

Table 14. XPRFEXIT return codes

Return code (decimal)	Explanation
0	All actions within the exit completed successfully.
4	Some or all actions finished with warning.
>=8	Some or all actions finished with error.

Return codes 1, 2, 3, 5, 6, 7, and less than 0 are reserved for IBM use only.

Return Code Handling

IBM Wave cannot initialize the service machine server code if the XPRFEXIT exit completes with a return code greater than or equal to 8. For more information, see Table 15 on page 251:

Table 15. XPRFEXIT return code handling

Return code (decimal)	Handling
0	IBM Wave considers the actions in the exit to be finished successfully and proceeds to initialize the IBM Wave service machine server code.
4	IBM Wave considers the actions in the exit to be finished with a warning and proceeds to initialize the IBM Wave service machine server code as normal.
>=8	IBM Wave considers the actions in the exit to be finished with an error and the IBM Wave service machine server code cannot be initialized.

Example

You can copy Figure 82 on page 252, and the modify it as necessary.

```
/* Sample user-supplied exit for service machines' PROFILE EXEC */
Trace 'Off'
ADDRESS CMS

devno = 399
'PIPE',
'   CP Q V' devno,
'|  SPECS W5 1',
'|  VAR accmode'

SELECT
WHEN accmode == "not" THEN
  DO
    message = 'Invalid device number:' devno '- unable to continue'
    returnCode = 8
  END

  WHEN accmode == "R/O" THEN
    DO
      message = 'WARNING! Device' devno 'accessed as R/O'
      returnCode = 4
    END

  OTHERWISE
      message = 'Device' devno 'accessed as R/W'
      returnCode = 0
END

SAY message
'CP MSG OPERATOR' message
EXIT returnCode
```

Figure 82. REXX example for XPRFEXIT

XVDSKOFF - DASD volume OFFLINE exit

Use the XVDSK0FF exit to customize the way the IBM Wave varies a Direct Access Storage Device (DASD) volume offline. Typically, a **CP VARY OFFLINE** command is used. However, if your site uses Remote Copy technology between storage controllers, other relevant commands can be run by using the exit.

Point of processing

The XVDSK0FF exit is called to handle the DASD Volume offline-processing, by using the **Vary Offline** action. Access **Vary Offline** in the **Current System Viewer** > **Storage** > **Volumes** viewer (right-click on the volume, and the select **More Actions**).

Exit code location

The XVDSKOFF exit is a REXX exec that is stored in the 399 minidisk of the WAVEWRKS service machine. The file name for the REXX must be XVDSKOFF EXEC.

Parameters sent to the exit

The XVDSKOFF exit receives two parameters, which are separated by a comma (","):

Disk Name

The name of the DASD volume (VOLSER).

Disk Real Address

The real address of the DASD volume, in hexadecimal form.

Return Codes

The XVDSKOFF exit must return a valid return code as specified in Table 16 on page 252

Table 16. XVDSKOFF exit return codes

Return code (decimal)	Explanation
0	All actions within the exit completed successfully.
4	Some or all actions finished with warning.
>=8	Some or all actions finished with error.

Return Code handling

If the return code from the exit is greater than 4, IBM Wave does not mark the DASD Volume offline. Return code handling is shown in Table 17 on page 253.

Table 17. XVDSKOFF exit return code handling

Return code (decimal)	Handling
0	IBM Wave considers the actions in the exit to be finished successfully and marks the DASD Volume offline. The **`Vary Offline`** action is marked as completed successfully.
4	IBM Wave considers the actions in the exit to be finished with a warning. IBM Wave marks the DASD Volume offline, and presents the user with the output from the exit. The **`Vary Offline`** action finishes with a warning status.
>=8	IBM Wave considers the actions in the exit to be finished with an error. IBM Wave cannot mark the DASD Volume offline, and presents the user with the output from the exit. The **`Vary Offline`** action is marked as "Ended with error".

Example

Figure 83 on page 254 is an example for the XVDSKOFF user exit.

```
/* REXX */

/* Parse the arguments - DASD Volume name and real address */
PARSE ARG diskName','diskRealAddress

ADDRESS COMMAND

/* First, set the privclass */
SAY 'Setting Privileges...'
'PIPE CP SET PRIVCLASS * =ABCDEFG | SPECS PAD > 1-* 3 | CONSOLE'
SAY 'Query device '
'PIPE CP Query DASD 'diskName' | var dasddata '
if (RC <> 0) then do
   say "Disk "diskName" was not found, action aborted "
   exit 8;
end
say "Device information is "dasddata
parse upper var dasddata DASD rDEV data
if (substr(Strip(data,"B"),1,14) = "WAS NOT FOUND.") then do
   say "Disk "diskName" was not found, action aborted "
   exit 8;
end
if (rdev <> right(diskRealAddress,4,'0')) then do
   say "Disk "diskName" invalid rdevice, action aborted "
   exit 8;
end
if (substr(Strip(data,"B"),1,7) = "OFFLINE") then do
   say "Disk "diskName" already offline "
   exit 4;
end
if (substr(Strip(data,"B"),1,8) = "CP OWNED") then do
   say "Disk "diskName" is CP OWNED, action aborted "
   exit 8;
end
if (substr(Strip(data,"B"),1,11) = "ATTACHED TO") then do
   parse var data cp owned owner .
   say "Disk "diskName" is owned by "owner", action aborted "
   exit 8;
end

if (substr(Strip(data,"B"),1,9) = "CP SYSTEM") then do
   parse var data cp sys name used .
   if (used > 0) then do
      say "Disk "diskName" is in use, action aborted "
      exit 8;
   end
   /* Detach the volume from SYSTEM before varying offline */
   SAY 'Detaching DASD Volume 'diskName' from SYSTEM...'
  'PIPE CP DET 'diskRealAddress' FROM SYSTEM | ',
      'SPECS PAD > 1-* 3 | CONSOLE'
   SAY '>>RC from detach: 'RC
   SAY ''
end

/* If the detach was successful, or if the DASD Volume */
/* was not attached in the first place, issue VARY OFFLINE */
if RC = '0' | RC = '121' then do
  SAY 'Varying device 'diskRealAddress' offline...'
  'PIPE CP VARY OFFLINE ' diskRealAddress' | ',
    'SPECS PAD > 1-* 3 | CONSOLE'
  SAY '>>RC from vary offline: 'RC
  SAY ''
end

/* Return the RC (either from the detach or from the VARY OFFLINE */
SAY 'Final RC: 'RC
Return RC
```

Figure 83. REXX example for the XVDSKOFF user exit

XVDSKON - DASD volume ONLINE exit

Use the XVDSKON exit to customize the way the IBM Wave varies a DASD volume online. Typically, doing so requires the **CP VARY ONLINE** command. However, if the site utilizes REMOTE COPY between storage controllers, other relevant commands can be issued when using the XVDSKON exit.

Point of processing

The XVDSKON exit is called to handle the actual DASD Volume online-processing, using the **Vary Online** action that is accessible from the DASD Volume popup menu.

Exit code location

The XVDSKON exit is a REXX exec stored on the 399 minidisk of the WAVEWRKS service machine. The file name for the REXX exec must be XVDSKON EXEC.

Parameters sent to the exit

The XVDSKON exit receives two parameters, separated by a comma (","):

Disk Name

The name of the DASD Volume (VOLSER).

Disk Real Address

The real address of the DASD Volume, in hexadecimal form.

Return Codes

The XVDSKON exit must return a valid return code as specified in Table 18 on page 255:

Table 18. XVDSKON exit return codes

Return code (decimal)	Explanation
0	All actions within the exit have completed successfully.
4	Some or all actions finished with warning.
>=8	Some or all actions finished with error.

Return Code Handling

IBM Wave does not mark the DASD Volume online if the return code from the exit is greater than 4. Detailed handling is shown in Table 19 on page 255:

Table 19. XVDSKON exit return code handling

Return code (decimal)	Handling
0	IBM Wave will consider the actions in the exit to have finished successfully and will mark the DASD Volume online. The "Vary Online" action will be marked as completed successfully.
4	IBM Wave will consider the actions in the exit to have finished with a warning. IBM Wave will mark the DASD Volume online, and will present the user with the output from the exit. The "Vary Online" action will finish with a warning status.
>=8	IBM Wave will consider the actions in the exit to have finished with an error. IBM Wave will not mark the DASD Volume online, and will present the user with the output from the exit. The "Vary Online" action will be marked as "Ended with error".

Example

The following example is a REXX example for the XVDSKON user exit.

```
/* REXX */

/* Parse the arguments - DASD Volume name and real address */
PARSE ARG diskName','diskRealAddress

ADDRESS COMMAND

/* First, set the privclass */
SAY 'Setting Privileges...'
'PIPE CP SET PRIVCLASS * =ABCDEFG | SPECS PAD > 1-* 3 |
CONSOLE'
SAY ''

/* Vary the device online */
SAY 'Varying device 'diskRealAddress' online...'
'PIPE CP VARY ONLINE 'diskRealAddress' | SPECS PAD > 1-* 3 | CONSOLE'
SAY '>>RC from Vary online: 'RC
SAY ''

/* If the DASD Volume was varied online successfully, */
/* Attach it to SYSTEM                                */
if RC = '0' then do
  SAY 'Attaching device 'diskRealAddress' to SYSTEM...'
  'PIPE CP ATTACH 'diskRealAddress' TO SYSTEM |',
    'SPECS PAD > 1-* 3 | CONSOLE'
  SAY '>>RC From attach: 'RC
  SAY ''
end

/* Return the RC (either from VARY ONLINE or from ATTACH) */
SAY 'Final RC: 'RC
Return RC
```

Figure 84. REXX example for the XVDSKON user exit.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States (www.ibm.com/legal/us/en/copytrade.shtml).

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Online Privacy Statement Highlights at http://www.ibm.com/privacy/us/en/ and the IBM Online Privacy Statement at http://www.ibm.com/privacy/details/us/en/ in the section entitled "Cookies, Web Beacons and Other Technologies", and the IBM Software Products and Software-as-a-Service Privacy Statement at http://www.ibm.com/software/info/product-privacy.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

Product Number: 5648-AE1

Printed in USA

SC27-6118-14