

```

1 -----*.*.*.*.*.General Info *.*.*.*.*-----
2
3
4 GIservr 6.0 requires Java 1.5 or higher on *nix systems. and Java 1.8 or higher
5 on Windows based platforms.
6
7 `lserv` is no longer required to run GIservr on *nix systems.
8
9 Passwords are no longer stored in the user's *.cfg file. Tools have been
10 provided to make changes to user accounts.
11
12 Many configuration variables have been deprecated. Please check the
13 configuration settings below to see what you may be able to remove from your
14 config files.
15
16 Heartbeat messages are no longer required for GIservr 6.0.
17
18 GIservr 6.0 has moved to a NIO back-end, you should see moderate speed
19 increases in performance.
20
21 Scripts provided by GIservr 6.0
22   % giservr - The main application script.
23   % giadmin - Remote administration module for the command line.
24   % giaccount - Local administration module for on system changes.
25   %
26   %
27 giservr
28   % This script executes the application loader for GIservr, it can be used to
29   % create service scripts for various platforms.
30   %
31   % To start GIservr execute `./giservr&`
32   %
33 giaccount
34   % USAGE giaccount command [args]
35   % -p | passwd: Change a user's password.
36   % giaccount -p username (new_pass | current_pass new_pass)
37   % Requires current password if not root.
38   % -i | invalidate: Invalidate a user's password.
39   % giaccount -i username
40   % -a | useradd: Add a new user.
41   % giaccount -a username password
42   % -d | userdel: Delete a user.
43   % giaccount -d username
44   % help: Display this help message
45   %
46   % This is the local administration tool, does not require passwords if logged
47   % in as root.
48   %
49 giadmin
50   % Remote administration shell.
51   % Enter server number or 'M' to manually enter information when prompted.
52   %
53   % Available Commands
54   % quit: Exit the interactive shell.
55   %
56   % exit: Exit the interactive shell.
57   %
58   % useradd: Add a new user.
59   %
60   % userdel: Delete an existing user.
61   %
62   % passwd: Change a user's password.
63   %
64   % list: List all users.
65   %
66   % active: List all active users.
67   %
68   % count: Display count of active users.
69   %

```

```

70  --invalidates: Invalidate a user's password.
71
72
73  ----- * * * * * Configuration * * * * * -----
74
75  Logging:
76  --logLevel=(off|severe|warning|info|config|fine|finer|finest), default info
77  -- Sets logging level from least->most verbose.
78
79  --clientLogToServer=(true|false), default false
80  -- Sets if client related logs get their own files or not.
81
82  --logConenctions=(true|false), default false
83  -- Sets if connection events should be logged.
84
85  --logPrograms=(true|false), default false
86  -- Sets if program events should be logged.
87
88  --logMemory=(true|false), default false
89  -- Sets if memory events should be logged.
90
91  --logAdmin=(true|false), default false
92  -- Sets if admin interface events should be logged.
93
94  --debug=(true|false), default false
95  -- Sets if debug events should be logged (sets logLevel to fine).
96
97  --log=(true|false), default false
98  -- Sets if any events should be logged.
99
100 --logSize=(integer), default 0 (infinite)
101 -- Sets the maximum log size before rotation in bytes (cumulative).
102
103 --logSizeKB=(integer), default 0 (infinite)
104 -- Sets the maximum log size before rotation in kilobytes (cumulative).
105
106 --logSizeMB=(integer), default 5
107 -- Sets the maximum log size before rotation in megabytes (cumulative).
108
109 --maxLogRotations=(integer), default 5
110 -- Sets the maximum number of log rotations to keep (0 is infinite).
111
112
113  General Configuration:
114  --useActiveDirectory=(true|false), default false
115  -- Sets if the server should authenticate using Active Directory or built
116  -- in authentication.
117
118  --activeDirectoryDomain=(string), default is Undefined
119  -- Sets the active directory domain for authentication, e.g. my_domain.com.
120
121  --activeDirectoryServer=(string), default is Undefined
122  -- Sets the active directory server for authentication,
123  -- e.g. local.my_domain.com or the IP address.
124
125  --password=(string), default is Undefined
126  -- Sets the password for the admin account.
127
128  --serverName=(string), default is from system
129  -- Sets the name of the server.
130
131  --serverAddress=(string), default is from system
132  -- Sets the address for the server to listen on.
133
134  --adminName=(string), default admin
135  -- Sets the username for the admin account.
136
137  --idleout=(integer), default 0 (never)
138  -- Sets how long before a session is automatically disconnected due to

```

```

139     being idle (in seconds).
140     """
141     deadlockDetectionSeconds=(integer), default 10
142     """ Sets how long before a session is considered deadlocked when reading and
143     writing becomes blocked.
144     """
145     expireDays=(integer), default 0
146     """ Sets how many days until a password is marked as expired.
147     """
148     port=(short), default 4350
149     """ Sets the port the server listens on for normal requests.
150     """
151     adminPort=(short), default 4450
152     """ Sets the port the admin interface can use to connect.
153     """
154     admin=(true|false), default true
155     """ Sets whether the admin account is active for this server.
156     """
157     remoteShutdown=(true|false), default false
158     """ Sets if remote shutdowns are allowed from the administrative interface.
159     """
160     maxLoginAttempts=(integer), default 0
161     """ Sets how many failed logins (GIclient) before automatic disconnect.
162     """
163     doesNotExpire=(true|false), default false
164     """ Sets if passwords expire or not (Can be set in the user's *.cfg file)
165     """
166     exec=(string), default fpdaemon
167     """ Sets the path to the fpdaemon program for the server to use.
168     """
169
170 Password Configuration:
171     minimumLength=(integer), default 5
172     """ Sets minimum length of a password.
173     """
174     maximumLength=(integer), default 48
175     """ Sets maximum length of a password.
176     """
177     expireDays=(integer), default 0 (does not expire)
178     """ Sets how many days until a password is marked as expired. Overrides
179     general configuration.
180     """
181     reuseDays=(integer), default 0
182     """ Sets how many days until a password can be reused.
183     """
184     reuseInstances=(integer), 3
185     """ Sets how many password changes until a password can be reused.
186     """
187     requireUpper=(true|false), default false
188     """ Sets if a password requires at least one uppercase letter.
189     """
190     requireLower=(true|false), default false
191     """ Sets if a password requires at least one lowercase letter.
192     """
193     requireNumeric=(true|false), default false
194     """ Sets if a password requires at least one number.
195     """
196     requireSymbol=(true|false), default false
197     """ Sets if a password requires at least one symbol.
198     """
199     requireInitialChange=(true|false), default false
200     """ Sets if a password requires an initial change after upgrading/adding a
201     user.
202     """
203     complexityErrorMsg=(string)
204     """ defaults to "Entered Password is not sufficiently complex."
205     """ Sets the error message to display on a complexity check failure.
206     """
207     reuseErrorMessage=(string)

```

208 defaults to "Too soon to reuse this password."
209 Sets the error message to display on a reuse failure.
210