

# INDEX

## A

- access methods
  - cluster, 123–128
  - resource, 131–139
  - RMAPI, 121
- adoption rates, 1
- agents, 33–34. *See also* monitoring
  - binary-only versions of, 60
  - creating with GDS, 40–42, 59
  - deploying, 278–282
  - packaging versus not packaging, 278–279
  - shell programs for, 60–61
- apachect1 program, 69
- Apache Web servers
  - configuring, 39–40, 41
  - creating and activating resources for, 40–42
  - fault probing and, 43–44
  - preparing for SunPlex, 64–65
  - preparing with GDS, 38–40
  - starting/stopping, 39–40, 43
  - SunPlex Agent Builder and, 63–88
- application (App) servers, 323
- application management, 54
- applications, 160–161
  - bounded recovery time of, 47
  - client recovery and, 50–51
  - cluster awareness of, 52–53
  - controlling with DSDL, 185–192
  - crash tolerance of, 47
  - data service access and, 46
  - dependencies of, 291–292
  - developing cluster-aware, 321–329
  - developing for existing, 205–282
  - file location independence in, 47–49
  - information gathering on, 54–58
  - internode communication and, 297–299
  - logical hosts and, 49
  - logical interfaces and, 50
  - making cluster-aware, 33
  - monitoring, 33–34, 56–58
  - monitoring with callback methods, 97–100
  - monitoring with PMF, 157
  - multi-homed hosts and, 50
  - multiple instances of, 295
  - qualifying for high availability, 45–51, 294–295
  - qualifying for scalability, 294–296
  - remote shared memory and, 303
  - simulating failovers of, 114–116
  - simulating restart of, 113–114
  - starting, 54–55
  - stopping, 56

- application specific integrated circuits (ASIC), 1
- arrays
  - pointers versus, 318
- Art and Science of C, The* (Roberts), 167
- ASIC (application specific integrated circuits), 1
- availability, 1. *See also* high availability
  - application, 12–14
  - calculating, 13–14
  - cluster, 2–9, 14
  - cost and, 12–13, 14–18
  - downtime causes and, 14–18
  - measuring, 10–18
  - scalability and, 285
  - service, 10–12
- B**
- bin directory, 215, 217
  - creating, 78
- BOOT callback method, 95, 221, 222
  - idempotency of, 97
- bounded recovery time, 47, 208, 211
- Bourne shell, 64
- C**
- caching, global filesystems and, 26
- callback methods, 331
  - administration, 95
  - application monitoring with, 97–100, 157
  - BOOT, 95, 97, 221, 222
  - for CUPS, 254–278
  - definitions of, 93–100
  - developing, 226–278
  - exit codes of, 97
  - FINI, 95, 97, 116–117, 221, 222
  - helper shell functions and, 226–228
  - idempotency and, 96–97
  - INIT, 94, 97, 121, 122
  - initialization, 93–95
  - model of, 89–91
  - MONITOR\_CHECK, 96, 99–100, 247–249, 277–278
  - MONITOR\_START, 96, 99–100, 236–238, 263–265
  - MONITOR\_STOP, 96, 99–100, 238–239, 265–267
  - POSTNET\_STOP, 92–93, 94, 97, 221, 222
  - PRENET\_START, 92–93, 94, 221, 222
  - process management and, 153–166
  - reference guide to, 98
  - required, 92–93
  - RMAPI, 119–152
  - RTR file, 220–222
  - semantic model of, 90–91
  - START, 92–94, 229–232, 255–261, 257–261
  - STOP, 92–93, 94, 97, 232–236, 261–263
  - template for, 110–111
  - UPDATE, 95, 97, 249–251, 276–277
  - VALIDATE, 95, 227–228, 251–254, 271–276
- capacity, scalable services, 284
- CCR (cluster configuration repository), 32
- child levels, with PMF, 155
- child process monitoring, 158–159
- cl\_apid daemon, 324–325, 327–328
- clean shutdowns, 23
- cl\_eventd daemon, 324
- client recovery, 50–51, 210, 212–213
- client registration, 327
- CLIs. *See* command-line interfaces (CLIs)
- cluster access methods, 123–128
  - scha\_cluster\_close(), 124–126
  - scha\_cluster\_get(), 124, 126
  - scha\_cluster\_getlogfacility(), 126–127
  - scha\_cluster\_getnodename(), 127–128
  - scha\_cluster\_open(), 124
  - tags for, 123
- cluster awareness
  - application development with, 321–329
  - as application qualification, 52–53
  - CRNP and, 321, 322–329
  - non-cluster awareness versus, 321–322
  - scalability and, 296
- cluster compatible applications, 52, 53
- cluster configuration repository (CCR), 32
- cluster file system. *See* global file service (GFS)
- cluster membership monitor (CMM), 32
- cluster reconfiguration notification protocol (CRNP), 321, 322–329
  - client registration in, 327
  - client semantics in, 326–329
  - components of, 324–325
  - framework of, 323–324
  - message passing in, 327–329

- message types in, 352
  - resource properties, 351
  - resource types, 326
  - SC\_CALLBACK\_REG, 352
  - SC\_EVENT, 352
  - SC\_REPLY, 352
  - SUNW.Event resource type, 351
- clusters, 2–19
- availability of, 14, 15
  - compute, 4–5
  - configuration repository, 32
  - configuring for SunPlex Agent Builder, 64–65
  - example applications of, 7–9
  - GFS and FFS in, 28
  - grid computing, 7–9
  - high availability and, 2–9
  - load balancing and, 292–293
  - loosely coupled, 3–4
  - membership monitor, 32
  - networking, 33
  - nodes in, 22–23
  - purposes of, 4–7
  - quorum devices and, 32–33
  - state of, 52–53
  - tightly coupled, 4, 5
  - topologies for, 3–9, 6, 7
  - transport of, 33
- CMM (cluster membership monitor), 32
- command-line interfaces (CLIs)
- halockrun (1HA), 165–166
  - hatimerun (1HA), 164–165
  - pmfadm, 154–163
  - scdscreate (1HA), 213–216
- common UNIX print server (CUPS)
- application, 210–213
    - absence of ties to physical identity of nodes in, 212
  - bounded recovery in, 211
  - callback method creation for, 254–278
  - callback methods in, 221–222
  - client recovery and, 212–213
  - crash tolerance in, 211
  - cups\_probe, 263–265
  - data service access in, 211
  - extension properties of, 225–226
  - file location independence in, 211–212
  - logical interfaces and, 212
  - MONITOR\_CHECK callback method, 277–278
  - MONITOR\_START callback method, 263–265
  - MONITOR\_STOP callback method, 265–267
  - multi-homed hosts and, 212
  - probe, 267–271
  - probe loop in, 269–271
  - registering resource types in, 279–280
  - resource creation in, 280–281
  - resource management in, 282
  - resource type creation for, 216–218
  - retrieving properties of, 268
  - START callback method, 255–261
  - STOP callback method, 261–263
  - UPDATE callback method, 276–277
  - VALIDATE callback method, 271–276
- compute farms/grids, 3, 7–9
- clusters in, 4–5
- configuration
- Apache Web server, 39–40, 41
  - scdsconfig (1HA) for, 218–220
  - in SunPlex Agent Builder, 68–72
- costs
- of availability, 13–14, 14–18
  - of outages, 16–18
- C programming language
- CUPS callback methods in, 254–278
  - DSDL and, 167–204
  - generating with SunPlex Agent Builder, 74–75
  - resource development with, 60
  - RMAPI and, 119–120
  - RSMAPI, 303–316
  - SunPlex Agent Builder and, 64, 77
- C++ programming language, 61
- crash tolerance, 47, 207–208, 211
- CRNP. *See* cluster reconfiguration notification protocol (CRNP)
- CUPS application. *See* common UNIX print server (CUPS) application
- cups\_probe, 263–265
- D**
- data consistency, 295
- data service
- access to, 46, 207, 211
  - failover versus scalable, 53

- data service development library (DSDL),
    - 60, 167–204
    - application control and, 185–192
    - CUPS probe and, 267–271
    - environment management for, 172–174
    - extension properties, 345
    - fault monitoring and, 192–198
    - functions for resource groups in, 182–184, 347
    - functions for resources in, 174–182, 346
    - functions for resource types in, 184–185, 186, 348
    - functions in, 171–203
    - general utilities with, 198–203
    - handle initialization, 254–255
    - introduction to, 168–169
    - life cycle of, 171, 172
    - man page errors on, 172
    - memory handling and, 203
    - PMF program effects in, 348
    - program arguments in, 170–171
    - RMAPI versus, 185–186
    - RTR files and, 169
    - using, 169–171
  - data types, 146–152
    - enumerated, 146–147
    - structures, 148–152
  - debugging, with DSDL, 198–203
  - decide\_restart\_or\_failover function, 241–247
  - decision support systems (DSS), 12
  - DEMOhaaws package, removing old, 82–83
  - dependencies, scalable services and, 291–292
  - Designing Enterprise Solutions with Sun Cluster 3.0* (Elling, Read), 21
  - developing for existing applications, 205–282. *See also* common UNIX print server (CUPS) application; MySQL application
    - agent deployment and, 278–282
    - callback methods and, 220–222, 226–278
    - CUPS application, 210–213
    - example applications used, 206–213
    - extension properties and, 222–226
    - MySQL application, 206–210
    - packaging and, 278–279
    - project creation in, 213–226
    - resource group creation and, 280–281
    - resource management and, 282
    - resource type creation in, 214–218
    - resource type registration and, 279–280
    - RTR files and, 220–226
    - scdsconfig (1HA) in, 218–220
    - scdscreate (1HA) in, 213–214
    - system environment and, 205–206
  - development
    - callback method, 226–278
    - of cluster-aware applications, 321–329
    - developer responsibilities in, 294–296
    - for existing applications, 205–282
    - planning for, 45–61
    - of scalable services, 296–299
  - DID (disk IDs), global, 24
  - discrete queries
    - qualifying for high availability and, 46
  - disk IDs (DID), global, 24
  - disksets, creating named, 37
  - distributed lock manager (DLM), 289
  - distributed service mechanisms, 285–286
  - DLM (distributed lock manager), 289
  - DNS, 285–286
  - docs directory, 215, 217
  - document type definition (DTD), 325–326
  - downtime. *See also* availability
    - causes of, 14–18
    - planned, 11
  - DSDL. *See* data service development library (DSDL)
  - DSS (decision support systems), 12
  - DTD (document type definition), 325–326
- E**
- enterprise architecture, 54
  - enumerated data types, 146–147
  - environment variables, 158
  - err\_code*, 266–267
  - error handling
    - CRNP message, 328–329
    - DSDL, 198–203
    - remote shared memory and, 318–319
    - with *scha\_cluster\_close()*, 125–126
    - scha\_err\_t* in, 151
    - scha\_strerror()* and, 145–146
  - etc directory, 215, 216, 217
    - creating, 78

- execution hosts, 8–9
- extension properties, 222–226
- F**
- failover file services (FFS), 27–28
- failover resources, 28–29
- failovers
  - creating IPMP groups and, 41–42
  - data service, 53
  - decide\_restart\_or\_failover function, 241–247
  - global networking and, 290–291
  - resource types and, 67
  - scds\_failover\_rg, 182–183
  - simulating application, 114–116
  - Sun Cluster file services in, 27–28
  - Sun Cluster resources in, 28–29
  - versus scalability, 53, 67
- fault monitoring. *See also* monitoring
  - DSDL, 192–198
  - planning for, 54
  - with PMF, 161–162
- fault probing
  - callback methods for, 97–100
  - GDS and, 43–44
  - in SunPlex Agent Builder, 87–88
- fault recovery, planning for, 53
- fault tolerance, 18, 19
- FFS (failover file services), 27–28
- file location independence, 47–49, 208–209, 211–212
- file locking, 165–166, 297
- filesystems
  - creating global, 36–38
  - global, 25–27, 36–38
  - mounting, 38
  - network, 3, 8
- FINI callback method, 95, 221, 222
  - idempotency of, 97
  - removing resources with, 116–117
- G**
- GDS. *See* generic data service (GDS)
- generic data service (GDS), 35–44
  - agent creation with, 40–42
  - Apache Web server preparation in, 38–40
  - checking for installation of, 41
  - cluster preparation with, 36–40
  - fault probing and, 43–44
  - global filesystem preparation in, 36–38
  - limitations of, 44
  - operation of, 42–44
  - resource creation with, 59
  - scdscreate (1HA) and, 214
  - stopping, 43
- gethrtime()(3C), 269
- get methods
  - cluster access methods, 123–128
  - resource access methods, 131–135
  - resource group access methods, 136–139
  - resource type access methods, 128–131
  - RMAPI, 122–139
- GFS. *See* global file service (GFS); global filesystems (GFS)
- GID (group ID), 308
- GIF (global interface), 29
- GIN (global interface nodes), 290–291
- global device names, 37–38
- global devices, 23, 24–25
  - scalable services and, 288
- global file service (GFS)
  - scalable services and, 53, 288–289
  - sessionless services and, 296–297
- global filesystems (GFS), 25–27
  - preparing with GDS, 36–38
- global interface (GIF), 29
- global interface nodes (GIN), 290–291
- global networking, 289–291
  - scalable services and, 289–291
- grid computing. *See* compute farms/grids
- group ID (GID), 308
- H**
- halockrun (1HA), 165–166
- handle initialization, 254–255
- HARD\_TIMEOUT, 232
- HASStoragePlus, 27–28, 272–274
- hatimerun (1HA), 164–165
  - scds\_timerun and, 198
- high availability, 9–19
  - application availability and, 12–14
  - application qualification for, 45–51
  - application recovery time and, 12
  - the business case for, 10–18
  - cluster availability and, 14
  - definition of, 9
  - downtime causes and, 14–18
  - fault tolerance and, 18, 19

- high availability (*continued*)
  - with GDS, 35–44
  - remote shared memory and, 301–302, 318–319
  - service availability and, 10–12
- high-availability (HA) clusters, 5–7
  - application impact for node failures in, 7
  - availability and, 18–19
  - fault tolerance versus, 18–19
- high-performance clusters, 9
- high-performance computing (HPC), 3
  - non-networked scalable services versus, 286
- High Sierra filesystem (HSFS), 25
- horizontal scaling, 284
- hostname-port-protocol combinations, 175–177
- hostnames
  - determining logical IP addresses of node's, 298
  - retrieving lists of used, 180–182, 183–184
- hostnames variable, 84
- hot standby configurations, 5–6, 14
- HPC (high-performance computing), 3, 286
- httpd.conf file, 39–40
- HTTP over SSL (https), 293
- I**
- IBM DB2 Extended Enterprise Edition (EEE), 286
- idempotency
  - callback methods and, 96–97
  - definition of, 96
  - MONITOR\_STOP callback method, 238–239, 265–267
  - STOP callback method, 234, 261–263
  - UPDATE callback method, 249–251
- INADDR\_ANY, 49
  - multi-homed hosts and, 50
- index.html file, 39
- INIT callback method, 221, 222
  - definition of, 94
  - idempotency of, 97
- INSTALLDIR variable, 219
- Internet relay chat (IRC), 285–286
- Internet service providers (ISPs), 16
- internode communication, 297–299
- IP addresses
  - aliasing, 50
  - global networking and, 289–291
  - logical hosts and, 49
  - migrating, 46
- IPMP. *See* IP network multipathing (IPMP)
- IP network multipathing (IPMP), 32, 291
  - failover groups, creating, 41–42
- K**
- kill (1), 56
- Korn shell (ksh)
  - callback method creation and, 226–254
  - generating with SunPlex Agent Builder, 74–75
  - helper shell functions and, 226–228
  - resource development with, 60–61
  - SunPlex Agent Builder and, 64, 77
- L**
- LD\_PRELOAD, 51
- levels of nines, 12–13
- libdsdev.so library, 171
- libraries
  - C, 60
  - DSDL, 60, 167–204
  - libdsdev.so, 171
  - libscha, 120, 146–152
  - RSMAPI, 303–316
- libscha library, 120
  - data types in, 146–152
  - enumerated data types in, 146–147, 344
  - structures in, 148–152
- life cycles
  - barrier, 314
  - DSDL, 171–172
  - resource, 96–97
  - resource type, 100–101
- links, 208
- load balancing, 283
  - sticky, 292–293
  - weighted, 292, 293
  - wildcard sticky, 293–294
- logging
  - callback errors, 97
  - with DSDL, 198–203
  - event handling, 261

- logical hosts
  - application HA and, 49
  - resources for, 41, 210
- logical interfaces, 50, 210, 212
- LOGICAL\_PERNODE\_B3, 298
- loosely coupled clusters, 3–4
- M**
- make command, 278–279
- make(1) utility, 64
  - remaking packages with, 86–87
- man directory, creating, 78
- Mauro, Jim, 318
- McDougall, Richard, 318
- memory handling. *See also* remote shared memory (RSM)
  - with DSDL, 203
  - scds\_close and, 173–174
- memory segments
  - creating export, 306–307
  - identifying export, 309–311
  - importing, 311–313
  - locking, 319
  - publishing, 307–309
- Message Passing Interface (MPI), 9
- mirrored volumes, creating, 37–38
- MONITOR\_CHECK callback method, 96
  - CUPS application, 277–278
  - MySQL, 247–249
  - standby node status and, 99–100
- monitoring
  - applications, 33–34, 56–58
  - callback methods for, 97–100
  - child processes, 158–159
  - in CUPS, 263–265
  - fault, 54
  - MySQL probe program for, 239–247
  - probe loops in, 246–247
  - programs for, 56–58
  - retrieving probe information in, 243–245
  - in SunPlex Agent Builder, 70–71
  - Monitor\_retry\_count* extension property, 236–237
  - Monitor\_retry\_interval* extension property, 236–237
- MONITOR\_START callback method, 96
  - application monitoring with, 99–100
  - CUPS application, 263–265
  - MySQL, 236–238
- MONITOR\_STOP callback method, 96
  - application monitoring with, 99–100
  - CUPS application, 265–267
  - idempotency of, 97
  - MySQL, 238–239
- mount command, 25–26
- MPI (Message Passing Interface), 9
- multi-homed hosts, 50, 209, 212
- multiple instances, 295
- multiple processes, in SunPlex Agent Builder, 84–85
- mysqladmin command, 232
  - fault monitoring with, 161–162
- MySQL application, 206–210
  - bounded recovery in, 208
  - callback method creation in, 226–254
  - callback methods in, 221
  - client recovery and, 210
  - crash tolerance in, 207–208
  - data service access in, 207
  - decide\_restart\_or\_failover function, 241–247
  - extension properties of, 223–224
  - file location independence in, 208–209
  - helper shell functions in, 226–228
  - logical interfaces and, 210
  - MONITOR\_CHECK callback method, 247–249
  - MONITOR\_START callback method, 236–238
  - MONITOR\_STOP callback method, 238–239
  - multi-homed hosts and, 209
  - mysqlksh\_probe.ksh, 239–247
  - node identity binding and, 209
  - probe loop in, 246–247
  - probe program, 239–247
  - registering resource types in, 279–280
  - resource creation in, 280–281
  - resource management in, 282
  - resource type creation for, 214–216
  - restart\_service function, 239–241
  - retrieving extension properties in, 245–246
  - START method, 229–232
  - STOP callback method, 232–236
  - UPDATE callback method, 249–251
  - VALIDATE callback method, 251–254

**N**

NAFO (Network Adapter FailOver), 291

naming conventions  
   for global devices, 24  
   PMF nametag, 158

NAS (network attached storage), 8

net addresses, 175–177

Network Adapter FailOver (NAFO), 291

network attached storage (NAS), 8

network awareness  
   resource types and, 57, 67  
   in SunPlex Agent Builder, 76

network-aware resource types, 57

network filesystems (NFS), 3  
   in compute grids, 8

NFS (network filesystems), 3, 8

nodes. *See also* clusters  
   absence of ties to physical identity of, 22–23, 209, 212  
   definition of, 4  
   in global filesystems, 26–27  
   listing active, 299  
   not joined to clusters, 22–23

non-networked scalable services, 286

**O**

online transaction processing (OLTP), 12

open boot prom (OBP) prompt, 22–23

optags, 122

Oracle 9i Real Application Clusters (RAC), 52–53, 286

Oracle Parallel Server (OPS), 286

outages, costs of, 16–18

**P**

packages/packaging  
   developing for existing applications, 278–279  
   remaking with make(1) utility, 86–87  
   removing old DEMOhaaws, 82–83  
   versus not packaging, 278–279

page size, 305–306

parallel databases, 286

parse\_args function, 227, 228

peer-to-peer model, 298

performance, global filesystems and, 27

Perl programming language, 61

persistent networks  
   qualifying for high availability and, 46

pkgadd (1M) utility, 278–279

pkg directory, creating, 78

planning for development, 45–61  
   application information gathering in, 54–58  
   application qualification in, 45–51  
   getting around requirements and, 51–52  
   importance of, 45  
   programming language selection in, 60–61  
   scope determination in, 52–54  
   tool selection in, 58–60

pmfadm (1HA) command, 134, 154–163  
   UPDATE callback method and, 249, 250

PMF\_TAG, 237–238

pointers  
   arrays versus, 318  
   data structure and, 317–318

poll (2), 315

Port\_List, 44

port lists, 177–178

POSTNET\_STOP callback method, 92–93, 221, 222  
   definition of, 94  
   idempotency of, 97

PRENET\_START callback method, 92–93, 221, 222  
   definition of, 94

private networks  
   HA clusters and, 6  
   tightly coupled clusters and, 4

PROBE command  
   SunPlex Agent Builder, 70–71

Probe\_Command, 44

probes  
   CUPS, 267–271

process management, 153–166  
   file locking and, 165–166  
   halockrun (1HA) for, 165–166  
   hatimerun (1HA) for, 164–165  
   process monitor facility for, 154–163  
   time-based, 164–165

process monitor facility (PMF), 134, 154–163  
   action programs in, 159–160  
   application monitoring with, 157  
   child process monitoring with, 158–159  
   considerations with, 154–157

- fault monitoring in, 161–162
    - management facilities in, 349
    - method resource parameter collection for, 229–230
    - metrics with, 155
    - nametag status in, 160–161
    - program\_type* values, 186–187, 189–190
    - starting process monitors with, 157–158
    - status in, 160–161
    - stopping, 162–163
  - Process Monitor Facility, 92
  - programming languages
    - for resource types, 60–61
    - selecting, 60–61
  - program\_type* values, 186–187, 189–190
  - prototype(4) file, 87
  - proxy filesystem (PXFS). *See* global file service (GFS)
  - public network management (PNM), 291
  - publish-and-subscribe model, 302
- Q**
- quorum devices, 32–33
- R**
- RAC (Real Application Clusters), 52–53, 286
  - RAID (redundant arrays of independent disks), 1
  - README.mysql\_ksh file, 215–216
  - README.cups file, 217–218
  - reboot, 116
  - redundancy
    - application monitoring and, 33–34
    - fault tolerance and, 18
    - in HA clusters, 5
  - redundant arrays of independent disks (RAID), 1
  - reliability, 1
    - in-house techniques for, 1–2
    - measuring, 13
  - remote connectivity, 207
  - remote shared memory (RSM), 298, 299, 301–319
    - barriers in, 313–314
    - basics of, 301–303
    - data structure issues in, 317–318
    - error handling and, 318–319
    - exporting memory in, 306–307
    - high availability and, 318–319
    - identifying segments in, 309–311
    - importing segments in, 311–313
    - nonmapping operations in, 315–316
    - publishing exported segments in, 307–309
    - RSM API and, 303–316
    - starting operations in, 303–306
    - synchronization in, 314–315
  - resource access methods, 131–135
    - scha\_resource\_close()*, 134–135
    - scha\_resource\_get()*, 131–132, 134–135
    - scha\_resource\_open()*, 131
    - tags for, 132–134
  - resource group access methods, 136–139
    - scha\_resourcegroup\_close()*, 136
    - scha\_resourcegroup\_get()*, 136, 138–139
    - scha\_resourcegroup\_open()*, 136
    - tags for, 137
  - resource group manager (RGM), 28, 89–117
    - administration methods in, 95
    - application monitoring methods in, 96, 97–100
    - callback definitions and, 93–100
    - callback method use in, 110–117
    - callback model in, 89–91
    - idempotency and, 96–97
    - initialization methods in, 93–95
    - naming restrictions with, 107
    - resource type creation in, 101–109
    - resource type life cycles and, 100–101
    - RMAPI and, 120
    - scalable services and, 287–288
    - scalable services resources and, 287–288
    - semantic model of, 90–91
    - state versus status and, 142–144
    - Sun Cluster API callback methods and, 92–93
    - value restrictions with, 109
  - resource groups, 30–31, 108, 111
    - bring online, 112
    - creating, 280–281
    - DSDL functions for, 182–184
    - properties of, 334–335
    - state versus status and, 142–144
  - resource management, 282

- resource management API (RMAPI), 60, 119–152
  - access methods, 121, 337
  - cluster access methods, 123–128
  - cluster tags, 338
  - data types in, 146–152
  - DSDL and, 168
  - DSDL compared with, 185–186
  - get methods, 122–139
  - getting started with, 119–120
  - libscha library, 120
  - libscha library enumerated types, 344
  - limitations of, 168
  - overview of, 120
  - resource access methods, 131–135
  - resource group access methods, 136–139
  - resource group tags, 342
  - resource tags, 340–341
  - resource type access methods, 128–131
  - resource type tags, 339
  - SCHA control tags, 343
  - state versus status and, 142–144
  - utility methods, 139–146
- resource properties, 105–109, 333–334
  - attributes of, 109, 335
  - DSDL functions, 179, 180
  - group, 108
  - naming restrictions for, 107
  - scalable services resources and, 287–288
  - tunable, 107
  - types of, 105–109
  - updating, 113
  - for VALIDATE, 251–253
- resources. *See also* resource group manager (RGM)
  - anatomy of, 30–32
  - creating, 31–32, 111–112
  - creating and starting in SunPlex Agent Builder, 73–74
  - definition of, 30
  - deleting, 116–117
  - DSDL functions for, 174–182
  - failover, 28–29
  - failures to start, 73–74
  - getting names of, 178–179
  - idempotency and, 96–97
  - life cycles of, 96–97, 100–101
  - restarting, 179
  - scalable, 29
  - state versus status of, 142–144
  - stopping/removing, 74
  - switching nodes for, 112–113
  - types of, 30, 31
  - versioning, 100–101
- resource type access methods
  - scha\_resourcetype\_close(), 130–131
  - scha\_resourcetype\_get(), 128, 131
  - scha\_resourcetype\_open(), 128
  - tags for, 129
- resource type properties, 103–105, 332–333
  - CRNP, 326
- resource type registration (RTR) files
  - building, 102–105, 220–226
  - callback methods in, 220–222
  - creating, 101–102, 110
  - DSDL and, 169
  - example of, 102–105
  - extension properties and, 222–226, 223–226
  - process monitoring and, 158
  - resource property types and, 105–109
  - resource type properties and, 103–105
  - sections of, 101–102
  - SunPlex Agent Builder and, 59, 86
  - syntax rules for, 102
- resource types, 30, 31
  - building from scratch, 59–60
  - callback methods with, 110–117
  - changing, 85
  - CRNP, 326
  - configuring with SunPlex Agent Builder, 68–72
  - creating, 101–109
  - creating with scdscreate (1HA), 75–78, 213–216
  - creating with SunPlex Agent Builder, 65–68
  - DSDL functions for, 184–185, 186
  - failover versus scalable, 67
  - Generic Data Service (GDS), 35–44, 59
  - life cycles for, 100–101
  - network-aware, 57, 67
  - programming language selection for, 60–61, 68
  - properties of, 104–109
  - registering, 31–32, 41, 111, 279–280

- RTR file and, 101–105
  - for sessionless scalable services, 297
  - SunPlex Agent Builder for, 59, 63–88
  - tools for creating, 58–60
  - using already developed, 35
  - vendor and application names in, 66
- restarts
  - decide\_restart\_or\_failover function, 241–247
  - fault monitor process, 188
  - with PMF, 154–157
  - resource, 179
  - resource group, 184
  - restart\_service function, 239–241
  - scha\_control() for, 139–141
  - simulating, 113–114
- restart\_service function, 239–241
- retry count
  - with PMF, 155
- return on investment (ROI), 10
- RGM. *See* resource group manager (RGM)
- rgm/libdsdev.h, 171
- RMAPI. *See* resource management API (RMAPI)
- RSM. *See* remote shared memory (RSM)
- RSMAPI. *See* remote shared memory (RSM)
- rsm\_controller\_attr\_t data type, 305–306
- rsm\_get\_interconnect\_topology(3rsm), 303–304
- rsm\_memseg\_export\_create(3rsm), 306–307
- rsm\_memseg\_export\_publish(3rsm), 307–309
- rsm\_memseg\_import\_connect(3rsm), 311–313
- rsm\_memseg\_import\_map(3rsm), 311
- rsm\_memseg\_import\_put(3rsm), 315–316
- rsm.segmentid files, 310–311
- rtconfig file, 79–80, 85
- S**
- SAN (storage area networks), 8
- scalability
  - application requirements for, 51
  - availability versus, 4–5
  - clusters and, 5
  - failover versus, 53
  - scalable resources and, 29
- scalable coherent interconnect (SCI), 299, 301
- scalable services, 283–300
  - availability and, 285
  - benefits of, 284–286
  - definition of, 283
  - dependencies and, 291–292
  - developer responsibilities in, 294–296
  - developing new applications as, 296–299
  - fast failure recovery and, 285
  - global devices and, 288
  - global file service and, 288–289
  - global networking and, 289–291
  - internode communication and, 297–299
  - non-networked, 286
  - other distributed service mechanisms compared with, 285–286
  - parallel databases and, 286
  - public network management and, 291
  - remote shared memory and, 299
  - sessionless, 296–297
  - Sun Cluster facilities for, 287–294
- SC\_CALLBACK\_REG, 352
- scds\_close, 173–174
- scdsconfig (1HA), 78–83, 218–220
  - code and output of, 80–81
  - variables with, 219–220
- scdscreate (1HA), 75–78, 213–216
  - code and output of, 77–78
- scds\_error\_string, 199
- scds\_failover\_rg, 182–183
- scds\_fm\_action, 192–194
- scds\_fm\_print\_probes, 194
- scds\_fm\_sleep, 194–195
- scds\_fm\_tcp\_connect, 195
- scds\_fm\_tcp\_disconnect, 197
- scds\_fm\_tcp\_disconnect(), 260
- scds\_fm\_tcp\_read, 196
- scds\_fm\_tcp\_write, 196–197
- scds\_free\_ext\_property, 203
- scds\_free\_netaddr\_list, 203
- scds\_free\_net\_list, 203
- scds\_free\_port\_list, 203
- scds\_get\_ext\_confdir\_list, 175
- scds\_get\_ext\_monitor\_retry\_count, 175
- scds\_get\_ext\_monitor\_retry\_interval, 175
- scds\_get\_ext\_probe\_timeout, 175

- scds\_get\_ext\_property, 174–175
- scds\_get\_netaddr\_list, 175–177, 260
- scds\_get\_port\_list, 177–178
- scds\_get\_resource\_group\_name, 183
- scds\_get\_resource\_name\_, 178–179
- scds\_get\_resource\_type\_name, 184
- scds\_get\_rg\_\*, 184
- scds\_get\_rg\_hostnames, 183–184
- scds\_get\_rs\_\*, 179
- scds\_get\_rs\_hostnames\_, 180–182
- scds\_get\_rt\_\*, 184–185
- scds\_initialize, 172–173, 254–255
- scds\_pmf\_get\_status, 187–188
- scds\_pmf\_restart\_fm, 188
- scds\_pmf\_signal, 188–189
- scds\_pmf\_start, 189–190
- scds\_pmf\_stop, 190–191
- scds\_pmf\_stop\_monitoring, 191
- scds\_print\_netaddr\_list, 201
- scds\_print\_net\_list, 200–201
- scds\_print\_port\_list, 202
- scds\_restart\_resource, 179
- scds\_restart\_rg, 184
- scds\_simple\_probe, 197
- scds\_svc\_wait, 191–192
- scds\_syslog, 202, 261
- scds\_syslog\_debug, 202–203
- scds\_timerun, 198
- SC\_EVENT, 352
- scha\_cluster\_close(), 124–126
- scha\_cluster\_get(), 124, 125–126
  - CLI, 126
- scha\_cluster\_getlogfacility(), 126–127
- scha\_cluster\_getnodename(), 127–128
- scha\_cluster\_open(), 124
- scha\_control function
  - application monitoring with, 99–100
  - CLI, 141
  - simulating application failover with, 114–116
  - simulating application restart with, 113–114
  - tags for, 140
- scha\_err\_t, 151
- scha\_extprop\_value\_t, 149–151
- scha\_resource\_close(), 134–135
- scha\_resource\_get(), 131–132, 134–135
  - CLI, 135
- scha\_resourcegroup\_close(), 136
  - CLI, 138–139
- scha\_resourcegroup\_open(), 136
- scha\_resource\_open(), 131–132
- scha\_resource\_setstatus(), 141–144
  - CLI, 145
  - state versus status and, 142–144
- scha\_resourcetype\_close(), 130–131
- scha\_resourcetype\_get(), 128
  - CLI, 131
- scha\_status\_value\_t, 148
- scha\_str\_array\_t, 148–149
- scha\_strerror(), 145–146
- scha\_uint\_array\_t, 149
- scinstall (1m) utility, 64
- scinstall tool, resource creation with, 31–32
- SCI (scalable coherent interconnect), 299, 301
- SC\_REPLY, 352
- scrgadm command, 31–32, 41
- Search for Extra-Terrestrial Intelligence (SETI), 7–8
- segment IDs, 309–311
- SEND\_KILL, 236
- sessionless services, 296–297
- session state, 295
- shared disks, finding, 36–37
- shells
  - Bourne, 64
  - Bourne-Again, 64
  - Korn, 60–61, 74–75, 226–254
  - resource development with, 60–61
- SIGKILL
  - GDS and, 43
  - in MONITOR\_STOP callback method, 238–239
  - in MySQL, 232
  - process monitoring and, 164, 165
- SIGTERM, 43
- simple\_probe utility, 71
- single symmetric multiprocessor (SMP) systems, 284
- sleep 25 command, 156
- SMOOTH\_TIMEOUT, 232
- SMP (single symmetric multiprocessor) systems, 284
- socket connection, 207
- Solaris, 8, 38
  - starting applications in, 55

- Solaris Internals* (Mauro, McDougall), 318
- Solaris Operating Environment (SOE)
  - syslog facility, 125–126
- Solaris Volume Manager (SVM), 25, 36
- Solstice DiskSuite, 36
- SPARC, 64
- src directory, creating, 78
- START callback method, 92
  - CUPS application, 255–261
  - definition of, 93–94
  - method resource parameter collection for, 229–230
  - MySQL, 229–232
  - properties of, 257–258
  - syntax testing in, 231
  - verification of, 258–261
- start\_cmd\_args*, 231
- START command
  - multiple processes and, 84–85
  - SunPlex Agent Builder, 69
- Start Timeout field, 69
- state
  - cluster awareness and, 322–323
  - remote shared memory and, 302
  - status versus, 142–144
- status
  - process monitor, 160–161
  - state versus, 142–144
  - UPDATE method and, 251
- sticky load balancing, 292–293. *See also* load balancing
- STOP callback method, 92
  - CUPS application, 261–263
  - definition of, 94
  - idempotency of, 97
  - method resource parameter collection in, 232, 233–234
  - MySQL, 232–236
- stop\_cmd\_args*, 234
- STOP command, SunPlex Agent Builder, 69–70
- storage
  - crash tolerance and, 47
  - file location independence and, 47–49
  - global devices and, 23, 24–25
  - persistent, 47
  - in Sun Cluster, 23
- storage area networks (SAN), 8
- submit hosts, 8–9
- Sun Cluster
  - API callback methods in, 92–117
  - application monitoring in, 33–34
  - cluster nodes and, 22–23
  - failover file services in, 27–28
  - failover resources in, 28–29
  - failures and, 23
  - feature set of, 21–23
  - framework of, 23–34
  - getting started with, 35–44
  - global devices in, 24–25
  - global filesystems in, 25–27
  - high-availability features of, 2
  - installing, 22
  - resource anatomy in, 30–32
  - resource group manager in, 28, 89–117
  - scalable resources in, 29–30
  - scalable services facilities in, 287–294
  - storage in, 23
  - subtopics in, 32–33
- SunFire Link, 301
- Sun Grid Engine (SGE), 9
- SunPlex Agent Builder, 63–88
  - advanced fault probing in, 87–88
  - advanced topics in, 83–88
  - agent installation/control in, 72–75
  - changing resource types in, 85
  - changing sources in, 86
  - cluster preparation and, 64–65
  - command-line tools in, 75–83
  - configuration command in, 71–72
  - configuration with, 65, 68–72
  - Creation command in, 68
  - failed resource installation in, 83
  - GUI use in, 65–72
  - hostnames variable in, 84
  - installing, 64
  - language selection in, 68
  - multiple processes in, 84–85
  - PROBE command in, 70–71
  - remaking the package in, 86–87
  - removing old DEMOhaaws package in, 82–83
  - resource creation with, 59
  - resource type configuration with, 78–83
  - resource type creation with, 65–68, 75–78
  - RTR file in, 86

SunPlex Agent Builder (*continued*)  
 START command in, 69  
 start/stop/probe sources in, 86  
 STOP command in, 69–70  
 Working Directory, 66–67

SUNWapchr, 38  
 SUNWapchu, 38  
 SUNW.Event resource type, 351  
 SUNWscdev, 119–120  
 SUNWscgds, 72–73  
 SUNWscman, 119–120  
 switchover, callback methods for, 116  
 symbolic links, 48, 49, 51  
 “syncdir” option, 289  
 synchronization, RSM API, 314–315  
 syseventd(1M) daemon, 324  
 SYSLOG\_FACILITY, 251–252  
 system failure, 5

## T

TERM signals, stopping applications with, 56  
*thorough\_probe\_interval*, 269  
 tightly coupled clusters, 4  
 time-based process management, 164–165  
 timeouts, 56  
   in SunPlex Agent Builder, 79  
 time period, with PMF, 155–156  
 timestamps, 269  
 transaction processing (TP) monitors, 54  
 typedefs, void, 152

## U

UID (user ID), 308  
 UNIX. *See also* common UNIX print server (CUPS) application  
   filesystem (UFS), 25  
   global filesystems and, 25–27  
 UPDATE callback method, 95

CUPS application, 276–277  
 idempotency of, 97  
 MySQL, 249–251  
 uptime. *See* availability  
 user ID (UID), 308  
 /usr/cluster/include/scha\_tags.h file, 122  
 util directory, 215, 217–218  
   creating, 78  
 utility methods, 139–146  
   scha\_control(), 139–141  
   scha\_resource\_setstatus(), 141–144  
   scha\_strerror(), 145–146

## V

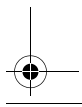
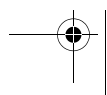
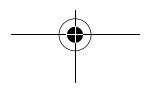
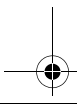
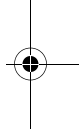
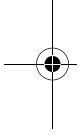
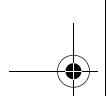
VALIDATE callback method, 95  
 CUPS application, 271–276  
 MySQL, 251–254  
   parsing function for, 227–228  
   resource commands for, 253–254  
   resource properties for, 251–253  
 Veritas filesystem (VxFS), 25  
 Veritas Volume Manager (VxVM), 25  
 version control, 213  
 virtual filesystems (VFS), 25  
 volume manager, 33

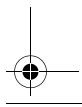
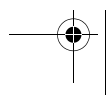
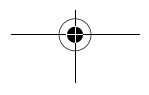
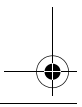
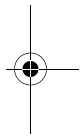
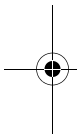
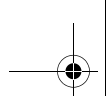
## W

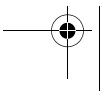
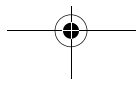
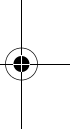
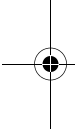
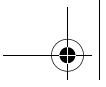
/web/httpd.conf file, 39  
 Web servers. *See* Apache Web servers  
 weighted load balancing, 292, 293. *See also*  
   load balancing  
 wildcards, 49, 50  
 wildcard sticky load balancing, 293–294.  
   *See also* load balancing  
 Working Directory, 66–67  
 wrapper scripts, 55

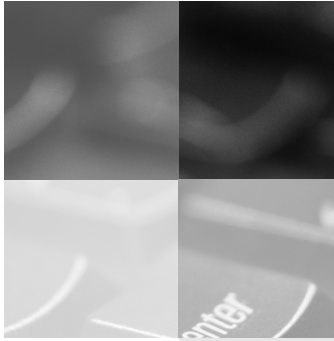
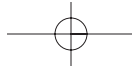
## X

XML document type definition (DTD), 325–326









# informIT

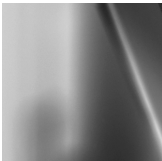
[www.informit.com](http://www.informit.com)

## YOUR GUIDE TO IT REFERENCE



### Articles

Keep your edge with thousands of free articles, in-depth features, interviews, and IT reference recommendations – all written by experts you know and trust.



### Online Books

Answers in an instant from **InformIT Online Book's** 600+ fully searchable on line books. For a limited time, you can get your first 14 days **free**.

POWERED BY  
**Safari**  
TECH BOOKS ONLINE™



### Catalog

Review online sample chapters, author biographies and customer rankings and choose exactly the right book from a selection of over 5,000 titles.



Wouldn't it be great  
if the world's leading technical  
publishers joined forces to deliver  
their best tech books in a common  
digital reference platform?

They have. Introducing  
**InformIT Online Books**  
powered by Safari.

■ **Specific answers to specific questions.**

InformIT Online Books' powerful search engine gives you relevance-ranked results in a matter of seconds.

■ **Immediate results.**

With InformIT Online Books, you can select the book you want and view the chapter or section you need immediately.

■ **Cut, paste and annotate.**

Paste code to save time and eliminate typographical errors. Make notes on the material you find useful and choose whether or not to share them with your work group.

■ **Customized for your enterprise.**

Customize a library for you, your department or your entire organization. You only pay for what you need.

**Get your first 14 days FREE!**

For a limited time, InformIT Online Books is offering its members a 10 book subscription risk-free for 14 days. Visit <http://www.informit.com/online-books> for details.



POWERED BY  
**Safari**  
TECH BOOKS ONLINE

**informIT**  
**Online Books**

**informit.com/onlinebooks**



Prentice Hall Professional Technical Reference

http://www.phptr.com/

Prentice Hall PTR InformIT InformIT Online Books Financial Times Prentice Hall ft.com PTG Interactive Reuters

PRENTICE HALL PTR

TOMORROW'S SOLUTIONS FOR TODAY'S PROFESSIONALS

Prentice Hall Professional Technical Reference

Browse Book Series What's New User Groups Alliances Special Sales Contact Us

Search | Help | Home

Quick Search

**PH PTR Online**

Keep Up to Date with

We strive to stay on the cutting edge of what's happening in professional computer science and engineering. Here's a bit of what you'll find when you stop by [www.phptr.com](http://www.phptr.com):

- ! What's new at PHPTR?** We don't just publish books for the professional community, we're a part of it. Check out our convention schedule, keep up with your favorite authors, and get the latest reviews and press releases on topics of interest to you.
- @ Special interest areas** offering our latest books, book series, features of the month, related links, and other useful information to help you get the job done.
- 👤 User Groups** Prentice Hall Professional Technical Reference's User Group Program helps volunteer, not-for-profit user groups provide their members with training and information about cutting-edge technology.
- ↔ Companion Websites** Our Companion Websites provide valuable solutions beyond the book. Here you can download the source code, get updates and corrections, chat with other users and the author about the book, or discover links to other websites on this topic.
- 📖 Need to find a bookstore?** Chances are, there's a bookseller near you that carries a broad selection of PTR titles. Locate a Magnet bookstore near you at [www.phptr.com](http://www.phptr.com).
- ✉ Subscribe today! Join PHPTR's monthly email newsletter!** Want to be kept up-to-date on your area of interest? Choose a targeted category on our website, and we'll keep you informed of the latest PHPTR products, author events, reviews and conferences in your interest area.

Visit our mailroom to subscribe today! [http://www.phptr.com/mail\\_lists](http://www.phptr.com/mail_lists)

Find a Bookstore

Book Series

Special Interests

Newsletters

Press Room

International

Best Sellers

Solutions Beyond the Book

Shopping Bag