# SmartFrog Quick Reference, version 3.18

```
Integer: 345
Long: 65325L or  65325l
Float: 34.76F or  34.76f or 34.76E-10F or 34.76e+10f or 34.76E10f
Double: 1534.45 or 1534.45D or 1534.45d or 1534.45E10 or 1534.45E-10D
String: "this is a string"
Binary data: #HEX#...#
Multi-line String: ## This is a string
                      Over many lines  #
Boolean: true or false
SFNull: NULL   // alternatively, leave the value empty
```

Operators: ! + - / == != >= <= < * ++ <> && ||

IF (val1 > val2) THEN (val1 – val2) ELSE (val2 – val1) FI;

Vectors:

```
v1  [1,2,3];
v2  [9,8,7];
v3  [v1, v2];        // same as [[1,2,3],[9,8,7]]
```

## Components

```
Component::= extends [LAZY] Type Body
Type::=    [ NULL | BaseReference ]
Body ::= (    ;   |  {  AttributeList }   )
```
The sfConfig attribute identifies the component in a description to deploy

```
sfConfig extends Prim {
    portNum 4047;
    sfProcessHost "ahost.smartfrog.org";
    administrators ["patrick",ATTRIB ROOT:sysadmin];
}
```

### Include files

Include files are resources on the classpath of the deploying JVM; they are expanded at parse time —before the components are deployed

```
#include "/org/smartfrog/services/jetty/components.sf"

system extends {
  #include "/org/smartfrog/services/filesystem/file.sf"
}
```

## References

```
Reference::= [LAZY] BaseReference
BaseReference::= ReferencePart ( : ReferencePart)*
```

| Reference Part | Meaning |
|----------------|---------|
| ROOT | root of the descriptor |
| PARENT | parent '..' |

| Reference Part | Meaning |
|----------------|---------|
| ATTRIB WORD | closest attribute 'WORD' |
| HERE WORD | local attribute 'WORD' only |
| WORD | ATTRIB if first/only part, else HERE |
| THIS | local component '.' |
| OPTIONAL(defval) | Gives an optional value for unresolved references |
| CONSTANT WORD \| STRING | Extracts static field from JAVA class (see LAZY) |
| PROPERTY WORD | JVM string property (see LAZY) |
| IPROPERTY WORD | JVM integer property (see LAZY) |
| ENVPROPERTY WORD | Environment variable; see LAZY) |
| IENVPROPERTY WORD | Environment variable; (see LAZY) |
| PROCESS | Current process |
| HOST WORD \| STRING | Host by reference/address |

LAZY: CONSTANT, PROPERTY, IPROPERTY, ENVPROPERTY, IENVPROPERTY and PROCESS references are evaluated in the JVM parsing the file, not the process hosting the components, *unless* LAZY is used to declare late binding of information.

## Functions

```
#include "org/smartfrog/functions.sf";
s extends formatString { format "path: $1"; -- ENVPROPERTY PATH;}
```
When a function is evaluated, its declaration is replaced by the result.

## Shell commands:

```
smartfrog -a action -f sfref -p port -e -d -headless -quietexit -?
sfRun -a sfref -e
sfPing hostname appname                   ping a host or process
sfList hostname                           list running applications
sfStart hostname appname URL              start an application
sfUpdate hostname appname URL             update a running application
sfTerminate hostname appname              terminate an application
sfStopDaemon hostname                     stop an application
sfDetachAndTerminate hostname appname     separate and stop an application
sfDetachAndTerminate hostname appname     separate and stop an application

sfParse [-v] [-q] [-r] [-R] {[-f] filename | sfref}
```
Action Descriptor : NAME : ACTION : SFREF : SUBREF : HOST : PROCESS

*sfref*: SmartFrog Resource Reference  —URL to the file; as a relative or absolute path name to a file, or as a path to a resource in a jar file on the classpath/current codebase.

## Well-Known Attributes

| | |
|---|---|
| sfConfig | the root of a SmartFrog description |

### Component

| | | |
|---|---|---|
| sfClass | class that implements a component | String |
| sfCodeBase | Java codebase for a component | String |
| sfDeployerClass | The class name for deployer | String |
| sfExport | enable/disable remote method calls | boolean |
| sfExportPort | fixes the port used by the component | Integer |
| sfSchemaDescription | the definition of a schema | CD |
| sfSyncTerminate | asynchronous or synchronous termination of children | boolean |

#### Location

| | | |
|---|---|---|
| sfProcessComponentName | name a component in root process | String |
| sfProcessHost | host where to deploy component | String |
| sfProcessName | subprocess where to deploy component | String |

#### Liveness:

| | | |
|---|---|---|
| sfLivenessDelay | how often to send liveness in seconds. | Integer |
| sfLivenessFactor | how many multiples of the liveness delay to wait till a liveness failure of the parent is declared | Integer |

#### SubProcesses:

| | | |
|---|---|---|
| sfProcessAllow | enable/disable subprocesses | boolean |
| sfProcessTimeout | Liveness timeout for subprocesses | Integer |
| sfRootLocatorPort | Registry port used by the rootProcess daemon. Default: 3800 | Integer |
| sfSubprocessGCTimeout | garbage collection timeout for subprocesses | Integer |
| sfProcessConfig | sets special attributes to next subprocess. | CD |
| sfProcessEnvVars | Sets environment variables | CD |

### Runtime attributes:

| | | |
|---|---|---|
| sfHost | The host address where a component runs | InetAddr |
| sfProcess | the process/subprocess name where a component runs. | String |

## Useful Components

| | |
|---|---|
| Prim | Primitive Component |
| Compound | Compound component. Shared lifecycle |
| ExecuteProgram, RunShellScripts, Java, Scripting | Run external programs, scripts, Java applications. |
| Downloader | Download a file from a URL |
| File, Mkdir, TextFile, TempFile | File manipulation components. |
| Trace | Tracing for life cycle hooks. see org/smartfrog/services/trace/components.sf |

## Management Console

sfManagementConsole -h <HOSTNAME> -p <PORT_NUMBER > -r -?

## Default configuration files for shell scripts

Ini file: dist/bin/default.ini or redefined using SFDEFAULTINI

Default deployment description: dist/bin/default.sf. Redefined using SFDEFAULTSF

Environment variables used:
    SmartFrog distribution directory: SFHOME
    User libs: SFUSERHOME,SFUSERHOME1, ..., SFUSERHOME4
    Remote class loading: SFCODEBASE, SFDYNAMICCLASSLOADING_ON
    Security: SFSECURITY,SFPRIVATE,SFHOSTNAME, SFSECURITY_ON