SSH Components

25 Oct 2007

SSH Components

25 Oct 2007

1 Introduction

The SSH component package, `sf-ssh`, provides SmartFrog components to connect to remote hosts, either to issue remote commands, or to copy files to and from the local host. This enables the components to be used to install SmartFrog itself on remote systems, or to manage a remote host that does not have SmartFrog itself installed. This can be useful when managing embedded systems, such as routers.

These components have been updated for SmartFrog 3.12.008; the changes are not yet complete. The recent and planned changes are both listed in this document.

2 Conceptual Model

SSH components are all derived from the `SSHComponent`. This is a workflow-enabled component: it can be configured to terminate after deployment, and so used in a workflow sequence.

The base component template defines common attributes for all components: the target host and port, a list of known hosts to trust, whether to trust all certificates,

```
SSHComponentSchema extends Schema {
    //authentication policy, one of "password" or "key"
    authentication extends String;

    //host to connect to
    host extends String;

    //should failures be reported
    failOnError extends OptionalBoolean;

    //for key authentication, the file containing the private key
    keyFile extends  OptionalFilenameType;

    //a list of known hosts; hosts to trust
    knownHosts extends OptionalVector;

    //a reference to a password provider
    passwordProvider extends Compulsory;

    //port to connect to
    port extends Integer;

    //timeout in milliseconds
    timeout extends Integer;

    //should all certificates be trusted
    trustAllCertificates extends Boolean;

    //the user name
    username extends String;
}


SSHComponent extends WorkflowPrim {
    sshSchema extends SSHComponentSchema ;
    authentication authenticate_password;
    failOnError true;
    port 22;
    sfShouldTerminate true;
    timeout 600000;
    trustAllCertificates true;
    //authentication policies
    authenticate_password "password";
    authenticate_key "key";
}
```

The default options of the `SSHComponent` trigger connections to port 22, the standard port for SSH, with trust for all remote certificates. The connection times out after ten minutes.

The authentication policy is either password or public key; public/private key connections are considered to be much more secure, though that does require a secure distribution and storage of the private keys.

Both policies also need a `passwordProvider`. This must be a LAZY reference to a component that provides passwords.

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

PasswordProvider extends InlinePassword {
    password "not-very-secret";
}
```

3 SSHExec Component

The `SSHExec` component can issue a series of commands to a remote server; the commands are represented as a list

```
SSHExec extends SSHComponent {
   sfClass "org.smartfrog.services.ssh.SSHExecImpl";
   sshSchema extends SSHComponentSchema  {
       commands extends OptionalVector;
       logFile extends OptionalString;
   }
   authentication SSHComponent:authenticate_password;
}


SSHExecWithPrivateKey extends SSHExec {
   authentication SSHComponent:authenticate_key;
}
```

Changes to SSHExec component in SmartFrog 3.12.008

- Choosing between authentication policies is a matter of switching authentication attribute, and providing the appropriate other attributes that specific authentication schemes may require
- the `passwordFile` attribute can be a reference to a `FileComponent`, as well as a path to a file.
- The attribute `sfShouldTerminate` is used to control whether the component should terminate or not.
- All validation of configuration values takes place before the connection is made

Planned Changes

Here are some items on the issue list for these components.

1. A planned change is to make this operation asynchronous, and to add an interface to allow other components to queue new commands.

4 ScpOperation

An `ScpOperation` is a component that can get or put multiple files in a single connection, over an SSH connection.

```
   ScpSchema extends SSHComponentSchema  {
       localFiles extends Vector;
       remoteFiles extends Vector;
       transferType extends String;
```

© Copyright 1998-2007 Hewlett-Packard Development Company, LP

```
    }

    /**
     * This is the basic Scp Operation
     */
    ScpOperation extends SSHComponent {
        sfClass "org.smartfrog.services.ssh.ScpComponentImpl";
        sshSchema extends ScpSchema;

        //get files from the remote server
        get_files "GET";
        //put files to the remote server
        put_files "PUT";

        //default transfer is a get
        transferType get_files;
    }

    /**
     * Switch to the public/private keypair
     */
    ScpOperationWithPrivateKey extends ScpOperation {
        authentication SSHComponent:authenticate_key;
    }
```

The list of local files and that of the remote files must be matched.

Changes to SCP component in SmartFrog 3.12.008

- Choosing between authentication policies is a matter of switching authentication attribute, and providing the appropriate other attributes that specific authentication schemes may require
- the `passwordFile` attribute can be a reference to a `FileComponent`, as well as a path to a file.
- The attribute `sfShouldTerminate` is used to control whether the component should terminate or not.
- All validation of scp operation parameters takes place before the connection is made
- Copies are made asynchronously. This helps for long-haul deployment, but means that you cannot rely on the copy to have completed before the next component in a simple compound is started. You must use workflow containers or otherwise register an interest in the event raised when the scp component is terminated.
- Liveness tests of the component now relay the state of the worker thread. If it fails, the liveness fails.

© Copyright 1998-2007 Hewlett-Packard Development Company, LP