

Configuring Internal Plug-Ins

Use the following configuration statements to configure internal plug-ins:

```
shared sae configuration plug-ins name name internal {  
    plug-in-class plug-in-class;  
}  
shared sae configuration plug-ins name name internal properties name {  
    value;  
}
```

To configure an internal plug-in:

1. From configuration mode, access the internal plug-in configuration.

```
user@host# edit shared sae configuration plug-ins name intnl internal
```

2. Configure the Java class name of the plug-in.

```
[edit shared sae configuration plug-ins name intnl internal]  
user@host# set plug-in-class plug-in-class
```

3. Access the internal plug-in property configuration.

```
[edit shared sae configuration plug-ins name intnl internal]  
user@host# edit properties name
```

For example, configure the classpath property, which is a list of additional libraries or paths for loading hosted plug-ins:

```
[edit shared sae configuration plug-ins name intnl internal]  
user@host# edit properties classpath
```

4. Configure properties that define the plug-in. Enter values for the property.

```
[edit shared sae configuration plug-ins name internalPlugin internal properties  
    name]  
user@host# set value
```

For example, enter values separated by commas for the classpath property where each value is a URL identifying one path or JAR file:

```
[edit shared sae configuration plug-ins name intnl internal classpath]  
user@host# set  
file:///opt/UMC/ssp/plugin/plugins.jar,http://plugins.com/ssp/plugins.jar
```

- Related Topics**
- Configuring Internal Plug-Ins (C-Web Interface)
 - Configuring the SAE for External Plug-Ins

- How Internal Plug-Ins Work
- Types of Internal Plug-Ins

Published: 2009-09-13