StreamServe™ Persuasion™

StreamServe Persuasion SP5 Command line utilities

User Guide

Rev B

StreamServe Persuasion SP5 Command line utilities User Guide
Rev B
© 2001-2010 STREAMSERVE, INC.
ALL RIGHTS RESERVED
United States patent #7,127,520

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of StreamServe, Inc. Information in this document is subject to change without notice. StreamServe Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book. All registered names, product names and trademarks of other companies mentioned in this documentation are used for identification purposes only and are acknowledged as property of the respective company. Companies, names and data used in examples in this document are fictitious unless otherwise noted.

StreamServe, Inc. offers no guarantees and assumes no responsibility or liability of any type with respect to third party products and services, including any liability resulting from incompatibility between the third party products and services and the products and services offered by StreamServe, Inc. By using StreamServe and the third party products mentioned in this document, you agree that you will not hold StreamServe, Inc. responsible or liable with respect to the third party products and services or seek to do so.

The trademarks, logos, and service marks in this document are the property of StreamServe, Inc. or other third parties. You are not permitted to use the marks without the prior written consent of StreamServe, Inc. or the third party that owns the marks.

Use of the StreamServe product with third party products not mentioned in this document is entirely at your own risk, also as regards the StreamServe products.

StreamServe Web Site
http://www.streamserve.com

Contents

Command line utilities

You can use the command line utilities instead of Control Center for example if:

- You can not connect to the server environment from a computer running Microsoft Windows.
- You want to automate or script certain tasks.
- You need to administer your server environment from within a UNIX environment

Prerequisites

The following must be installed:

- Framework and Control Center

About the utilities

The following utilities are covered in this documentation:

ss_territory	Create and administer application domains and StreamServe applications (Service Gateway, StreamStudio and StreamServer).
ss_scm	Create and administer StreamServer and service gateway services. It is not enough to create applications for StreamServers and service gateways. You must also create local services for these applications. **Note:** You do not need to create a local service for StreamStudio.
ss_deploy	Deploy a Design Center export file to a StreamServer application.
ss_rcp	Remote copy tool for copying files or folders to and from remote hosts, or for copying folders on a remote host.

Scenario 1 – Create application domain and run service

1. Use ss_territory to create an application domain.
2. Use ss_territory to create a StreamServer application in the application domain.
3. Use ss_scm to create a StreamServer service.
4. Use ss_deploy to deploy an export package.
5. Use ss_scm to start the StreamServer service.

Scenario 2 – Redeploy a StreamServer service

1. Use ss_scm to stop the StreamServer service
2. Use ss_deploy to deploy an export package.
3. Use ss_scm to start the StreamServer service.

Using the command line utilities

UNIX

1 Set the `$STRS_LOCATION` environment variable to where you installed the StreamServe software, for example:

```
STRS_LOCATION=/usr/streamserve/streamserve-5.5.0.GA.900
export STRS_LOCATION
```

2 Browse to the following directory:

```
$STRS_LOCATION/applications/managementgateway/bin
```

3 Run the following command:

```
./start <utility> <arguments>
```

For `ss_scm` arguments, see *Arguments for ss_scm utility* on page 8.

For `ss_territory` arguments, see *Arguments for ss_territory utility* on page 12.

For `ss_deploy` arguments, see *Arguments for ss_deploy utility* on page 19.

For examples, see *Examples* on page 23.

Windows

1 In the command line window, browse to the following directory:

```
<StreamServe_installation>\Platform\Core\<version>\bin
```

2 Run the following command:

```
<utility>.exe <arguments>
```

For `ss_scm` arguments, see *Arguments for ss_scm utility* on page 8.

For `ss_territory` arguments, see *Arguments for ss_territory utility* on page 12.

For `ss_deploy` arguments, see *Arguments for ss_deploy utility* on page 19.

For examples, see *Examples* on page 23.

Arguments for ss_scm utility

Required arguments

`-servicename <name>` where `<name>` is the service name.

`-action <action>` where `<action>` is one of the actions in the table below.

Note: The `-servicename <name>` argument is not required if `-action <action>` argument is `-action uninstall` or `-action printservices`

<action>	Description	Requires additional arguments
`start`	Starts a service.	
`stop`	Stops a service.	
`isrunning`	Checks whether a service is running or not.	
`pause`	Pauses a service	
`resume`	Resumes a service	
`newname`	Renames a service.	`-newname <name>`
`setstartuptype`	Changes the startup type of the services.	`-startuptype <startuptype>`
`setstartupuid`	Changes the server login user name. **Note:** A prompt "New password for service" is displayed where you specify the password for `-uid <username>`	`-uid <username>`
`setservicetype`	Sets the service type.	`-servicetype <type>`
`setdescription`	Sets the service description.	`-description <description>`

`create`	Creates a new service instance.	`-binpath <path>`
		`-description <description>`
		`-servicetype <servicetype>`
		`-startuptype <startuptype>`

`setbinpath`	Sets path to the service executable.	`-binpath <path>`
`getbinpath`	Gets path to the service executable.	

<action>	Description	Requires additional arguments
`delete`	Deletes the service. **Note:** This can corrupt the system if important services are removed	
`setarg`	Sets a startup argument **Note:** The argument must exist.	`-argname <argname>` `-argvalue <argvalue>`
`newarg`	Creates a new startup argument for specified service.	`-argname <argname>` `-argvalue <argvalue>`
`delsinglearg`	Delete a single startup argument e.g. `-demo`. (A single argument is an argument without a value)	`-argname <argname>`
`delbinaryarg`	Deletes a binary argument e.g. `-wd /home/user/projects/myproject/wd`	`-argname <argname>`
`getstartuptype`	Gets the service startup type.	
`getstartupuid`	Gets the service startup uid/username.	
`getservicetype`	Gets the service type.	
`getdescription`	Gets the service description.	
`printservices`	Gets installed services. If `-servicetype <servicetype>` is specified, only services of that type are returned.	
`printarguments`	Prints startup argument values for the service.	`-argname <argname>`
`getarg`	Gets startup argument value.	`-argname <argname>`
`uninstall`	Uninstalls all applications of the specified type. Also unregisters the applications from the application domain.	`-applicationtype <type>` `-applicationversion <version>` (Optional if action is uninstall)
`updateappbinpath`	Updates the executable path of all applications to match the path specified in the `-binpath` argument.	

Action dependent arguments	Description
`-newname <name>`	A new name of a service.
`-startuptype <type>`	The startup type can be one of: • `auto` • `manual` • `disabled`
`-uid <username>`	The user name to use for a service.
`-servicetype <type>`	The type can be one of: `STRSCS` – a StreamServer service `STRSDL` – a StreamStudio service. `STRSGW` – a service gateway service.
`-description <description>`	A description of the service.
`-binpath <path>`	Path to the StreamServer executable, for example: `/var/streamserve/streamserve-5.5.0.GA.370/applications/streamserve/start`
`-argname <name>`	The name of the startup argument.
`-argvalue <value>`	The value of the startup argument.
`-applicationtype <type>`	The type can be one of: `STRSCS` – a StreamServer application `STRSDL` – a StreamStudio application. `STRSGW` – a service gateway application. `STRSCI` – an Archiver application
`-applicationversion <version>`	The application version number.
`-repeat <number>`	Number of times to repeat the command. Default is `1`. Use only where it makes sense, for example with the `action -getarg -argname <argname>` argument.

Optional arguments	Description
`-host <hostname/ipnumber>`	If you do not specify this argument, localhost is used.

Optional arguments	Description
`-ipport <port>`	IP port number. If you do not specify this argument, `28000` is used.
`-timeout <ms>`	Timeout in milliseconds. If you do not specify this argument, `5000` ms is used.
`-cert <filename>`	The client certificate file. The path to the file must be included in the filename.
`-user <username>`	The user name for authentication. If you specify this argument, you must also specify the `-pass` argument.
`-pass <password>`	The password for authentication. If you specify this argument, you must also specify the `-user` argument.
`-v`	Output version and exit.
`-h`	Display the readme file. (This user guide covers the information in the readme file).

Arguments for ss_territory utility

Required argument

`-action <action>` where `<action>` is one of the following:

<action>	Description	Requires additional arguments
`get`	Gets the territory.xml for the specified application domain and creates a timestamp.	`-territoryid <territoryid>` or `-territoryname <territoryname>` `-filename <filename>`

`update`	Updates application domain information.
	By specifying the `-modtime` that was created when you used the get action, and the file has not been modified in the database since then, the update will succeed.
	By specifying `-force` instead of `-modtime`, the file will be overwritten without checking if it has been modified.
		`-territoryid <territoryid>` or `-territoryname <territoryname>`
		`-filename <filename>`
		`-force` (Required only if the last modification time is out of sync)
		Optionally you can specify `-modtime <modtime>` If the `-modtime` argument is not specified, the -force argument must be specified.

`delete`	Deletes an application domain	`-territoryid <territoryid>` or `-territoryname <territoryname>`

`create`	Creates an application domain	`-territoryid <territoryid>` or `-territoryname <territoryname>`
		`-filename <filename>`
		`-territoryversion <version>`
		`-territorydescription <description>`

`add_application`	Adds an application to the application domain	`-territoryid <territoryid>` or `-territoryname <territoryname>`
		`-appname <name>`
		`-apptype <type>`
		`-appvers <vers>`

`list_applications`	Lists applications in an application domain.	`-territoryid <territoryid>` or `-territoryname <territoryname>`

<action>	Description	Requires additional arguments
`del_application`	Deletes an application if it resides on the same machine as the management gateway. You can delete an application on a remote machine if you specify the application ID stored in the SER database.	`-appname <name>`
`rename_application`	Renames an application on the same machine as the management gateway. You must also run `ss_scm -action newname <name>` to synchronize the application name with the service name.	`-appname <name>`
`display_node_info`	Displays information on the current host	
`display_all_info`	Displays information on all hosts that are registered in the StreamServe Enterprise Repository.	
`display_all_domain_names`	Displays all application domains on the host.	`Optional:` `-territoryversion <version>`
`join`	Moves an application between application domains.	`-applicationid <id>` `-territoryid <id>`

`get_db_scripts`	Generates the database scripts for the specified application domain. The scripts are zipped in a tgz file and stored in `<StreamServe installation>/applications/managementgateway/etc` on Unix/Linux and `<StreamServe installation>\Applications\Management\<version>\etc` on Windows	`-territoryname <territoryname>`
		`-db_type <runtime/design>`
		If you specify `runtime`, scripts for the runtime database are created, if you specify `design`, scripts for the web content database are created.

<action>	Description	Requires additional arguments

`get_job_status`	Displays the job status, e.g. if a database was created with `create_appdomain_db`.
	When everything in a log has been retrieved by a client, this action with the same job ID will fail.
		`-jobid <jobid>`
		`-frompos <pos>`
		`-maxlen <max_bytes>`

`rename_territory`	Renames the application domain	`-territoryname <territoryname>`
`app_in_sync`	Checks if the `territory.xml` file for the application is in sync with the information in the StreamServe Enterprise Repository.	`-appname <name>`
`deploy_domain_info`	Exports the territory.xml to the specified application. You can only specify applications on the management gateway machine.	`-appname <name>`

`create_appdomain_db`	Creates a runtime database for the application domain.
	To check that the database was created, you must run the `get_job_status` action with the job ID that the `create_appdomain_db` action returned.
		`-dbuser <user>`
		`-dbpass <pass>`
		`-db_type <runtime/design>`
		If you specify `runtime`, scripts for the runtime database are created, if you specify `design`, scripts for the web content database are created.

`create_resource`	Creates a resource in the StreamServe Enterprise Repository. The resource can be a StreamStudio portal, an Archiver node, Document Type etc.	`-resource_version <version>`
		`-filename <path to resource file>`
		`-resource_name <name>`
		`-content_type <type>`
		`-strs_type <type>`
		Optional arguments:
		`-resource_description <description>`
		`-territoryid <ID>`
		`-nodeid <ID>`
		`-content_encoding <encoding>`

<action>	Description	Requires additional arguments
update_resource	Updates a properties on an existing resource. You can update the following properties: - Territory ID - Node ID - Resource name - Resource description - Content encoding - The resource data	-resource_id <*ID*>
assign_resource_to_territory_relation	Link a resource to an application domain. By using this action you can assign the same resource to several application domains.	-id <*resource_id*> -idto <*domain_id*>
remove_resource_to_territory_relation	Removes the link from the application domain to the resource.	-id <*resource_id*> -idto <*domain_id*>
remove_resource	Removes a resource from the StreamServe Enterprise Repository.	-resource_id <*ID*>
get_resource_data	Retrieves the resource data.	-resource_id <*ID*> -filename <*name*>
assign_resource_to_application_relation	Links a resource to an application. For example the Archiver application requires a link to the configuration resource.	-id <*resource_id*> -idto <*domain_id*>
remove_resource_to_application_relation	Removes the link from the application to the resource.	-id <*resource_id*> -idto <*domain_id*>
extract_document_types	Extracts the document types that are related to the applicaiton domain and saves the extracted data to a file.	-filename <*filename*> -territoryid <*ID*> *OR* -territoryname <*appdomain_name*>

<action>	Description	Requires additional arguments
`store_doctypes_and_update_db`	Stores document types and tries to update the runtime database. The action returns a job Id and creates a job in the management gateway. This job performs the update in the database, and you can use the job Id to retrieve the job status with the `-get_job_status` action.	`-territoryid <ID>` `-appname <name>`
`attach_ser_db`	Triggers the management gateway to load the `<ManagementGateway_root>\securityprofiles\enterpriserepository.xml` and connect to specified database.	

`list_hotfixes`	Lists hotfixes installed for the database specified by the `-db_type` argument.	`-dbuser <user>`
		`-dbpass <pass>`
		`-db_type <runtime/design/archive/enterprise>`
		`runtime` – the runtime database
		`design` – the web content database
		`archive` – StreamServe archive.
		`enterprise` – the StreamServe Enterprise Repository.

`db_apply_hotfixes`	Applies available hotfixes to the specified database.	`-dbuser <user>`
		`-dbpass <pass>`
		`-db_type <runtime/design>`
		If you specify `runtime`, hotfixes for the runtime database are applied, if you specify `design`, hotfixes for the web content database are applied.

Action dependent arguments	Description
`-territoryid <id>`	The application domain ID. Optionally use `-territoryname`
`-territoryname <name>`	The application domain name. Optionally use `-territoryid`

Action dependent arguments	Description
`-territoryversion <version>`	The application domain version.
`-territorydescription <description>`	An application domain description.
`-force`	Required for update action only if the last modification time is out of sync.
`-modtime <modtime>`	The last modification time returned by the `get` action.
`-applicationid <id>`	The application id.
`-filename <filename>`	The file name to use for the application domain configuration file.
`-appname <appname>`	The application name.
`-appvers <version>`	The application version.
`-apptype <type>`	The application type. The type can be one of: `STRSCS` – a StreamServer application `STRSDL` – a StreamStudio application. `STRSGW` – a service gateway application. `STRSCI` – an Archiver application.
`-jobid <jobid>`	The job ID.
`-dbuser <dbuser>`	Database administrator user.
`-dbpass <dbpass>`	Database administrator user password.
`-frompos <pos>`	Position in log file from which to retrieve content of the file. To get all content specify `0`.
`-maxlen <maxbytes>`	Max number of bytes to retrieve from the log file. To get all content specify a high number, e.g. `60000`.
`-resource_version <version>`	A string of your choice that represents the version of the resource, for example 1.0 or Alpha 0.1.
`-filename <path to resource file>`	The path to the file containing the resource data.
`-resource_name <name>`	The name of the resource.
`-content_type <type>`	A general content type, for example `application/x-streamserve.com-webportal`
`-strs_type <type>`	A specific content type, for example `application/x-streamserve.com-streamstudio`
`-resource_id <id>`	The resource ID given when creating the resource.

Action dependent arguments	Description
`-id <id>`	The resource ID when assigning and removing resources to and from domains and applications.
`-idto <domain_id>`	The application domain when assigning and removing resources to and from it.

Optional arguments	Description
`-resource_description <description>`	A description of the resource
`-territoryid <ID>`	Not used in this release.
`-nodeid <ID>`	The physical server hosting the resource.
`-content_encoding <encoding>`	Specifies if the data is compressed, for example `application/x-gzip`. Leave empty for plain data.

For more optional arguments, see *Arguments for ss_scm utility* on page 8.

Arguments for ss_deploy utility

Required arguments

`-action <action>` where `<action>` is one of the following:

Action	Description	Requires additional arguments
`deploy`	Deploys a package with a specified platform to an application.	`-physicalplatform <platform>` `-package <package>`
`deploy_application_config_files`	Use for Archiver and Service Gateway applications. This action requires that a working directory has been created for the application. The action copies all template files the application depends on to the working directory. For example, if you want application specific log settings, a specific logmanager.xml is required in the application's working directory. **Note:** This action only takes the `-appname` argument.	`-appname <name>`

Action dependent arguments

Argument	Description
`-appname <name>`	The StreamServer application to deploy to.
`-physicalplatform <physicalplatform>`	The physical platform to deploy. **Note:** A physical platform must exist in the export package.
`-package <package>`	The package file to deploy. The package file is the one exported from Design Center.

Optional arguments

Argument	Description
`-projectname <name>`	The name of the Project, normally the export file without the extension. If this argument is not specified the name of the exportfile is used.
`-projectlabel <label>`	The label to use, specify revision from CVS or other label.

For more optional arguments, see *Arguments for ss_scm utility* on page 8.

Arguments for ss_rcp utility

Required arguments	Description
`-src <source_name>`	The path to the file or folder to copy.
`-dst <name>`	The path to the file or folder to copy to.

Optional arguments	Description	Requires additional arguments
`-copydir`	Copies a folder recursively locally on the remote host where the management gateway is running. **Note:** On UNIX/Linux, any symbolic links and named pipes are also copied.	`-onremote`
`-R`	Copies a folder recursively to or from a remote host. **Note:** On UNIX/Linux, any symbolic links and named pipes are also copied.	
`-fromremote`	Copy a file from the remote host to the local host. If you do not specify `-fromremote`, a file is copied from the local host to the remote host.	
`-onremote <remotehost/ip>`	Copy locally on the remote host.	
`-host <hostname/ip>`	The remote host to copy to or from. By default, `localhost`.	
`-ipport <port>`	The IP port number to use. By default, `28000`.	
`-timeout <ms>`	Time-out in milliseconds. By default, `5000`.	

Optional arguments	Description	Requires additional arguments
`-startrange <pos>`	The position in the file where copying starts, specified in number of bytes from start of file. Ignored for `-R` or `-copydir`.	
`-stoprange <pos>`	The position in the file where copying stops, specified in number of bytes from start of file. Ignored for `-R` or `-copydir`.	
`-cert <filename>`	The client certificate file	
`-user <username`	The user name for authentication	`-pass`
`-pass <password`	The password for authentication	
`-unpack`	Only valid if copying a file to a remote host. The file is unpacked as a `.tgz` file.	
`-v`	Display version information and exit.	
`-h`	Display help and exit.	

Examples

The following examples are for Windows where the commands are run from `<StreamServe_installation>\Platform\Core\<version>\bin`

For UNIX, instead of running the `.exe` files, you browse to `$STRS_LOCATION/streamserve/applications/managementgateway/bin` and run for example `./start ss_territory <args>`

Creating an application domain

Prerequisites

You must have access to a `territory.xml` whose settings reflect your environment. You can for example use an existing `territory.xml` file and modify it according to your needs or create a new one from scratch.

Command

```
ss_territory.exe -user USERID -pass PASSWD -cert C:\Program
Files\StreamServe\Platform\Core\<version>\bin\security\certificat
estore\trusted\authorities\streamserve.ca.crt -action create -
territoryname MYAPPDOMAIN -filename "territory.xml"
-territoryversion 5.5.0 -territorydescription "My New Application
Domain"
```

Modifying an existing application domain

Prerequisites

You must have access to a `territory.xml` whose settings have been modified to reflect your environment. You can for example use an existing `territory.xml` file and modify it according to your needs or create a new one from scratch.

Command

```
ss_territory.exe -user USERID -pass PASSWD -cert C:\Program
Files\StreamServe\Platform\Core\<version>\bin\security\certificat
estore\trusted\authorities\streamserve.ca.crt -action update -
territoryname MYAPPDOMAIN -filename territory.xml -
territoryversion 5.5.0 -territorydescription "My Updated
Application Domain" -force
```

Creating a StreamServer application

Prerequisites

An application domain must be created.

Command

```
ss_territory.exe -user USERID -pass PASSWD -cert C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt -action add_application -
territoryname MYAPPDOMAIN -appname MYAPPLICATION -appvers 5.5.0 -
apptype STRSCS
```

Note: Make a note of the application Id that is returned, you will need it when creating the service.

Creating a service for a StreamServer application

Four commands are required as below, the first is to create the service and the other three to add arguments to the service, for it to be possible to start from command line.

Note: When you specify the service name, make sure it is the same name as the application you created with the `ss_territory` command.

Prerequisites

A StreamServer application must be created.

Commands

```
ss_scm.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\<version>\bin\security\certificat
estore\trusted\authorities\streamserve.ca.crt" -action create -
servicename MYSTRSSERVICE -startuptype manual -binpath "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\bootloader.exe" -
servicetype STRSCS -description "My Streamserver Service"
```

```
ss_scm -cert <path to cert> -action newarg -argname "-kernel"
-argvalue file://kernel_5_5_0.xml -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname "-env"
-argvalue "file://<path to the .environment file used by the
StreamServer>" -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname
"-applicationid" -argvalue "<applicationid>" -servicename
<servicename>
```

Deploying an export package to an application

Prerequisites

- A service for a StreamServer application must be created.
- A working directory containing an export package must exist.

Command

```
ss_deploy.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt" -action deploy -appname
MYSTRSSERVICE -physicalplatform MYPHYSICALPLATFORM -package
"MYCONFIGURATION.export" -projectlabel 1.0
```

```
ss_scm -cert <path to cert> -action newarg -argname "-kernel" -
argvalue file://kernel_5_5_0.xml -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname "-env" -
argvalue "file://<path to the .environment file used by the
StreamServer>" -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname "-
applicaitonid" -argvalue "<applicationid>" -servicename
<servicename>
```

Note: To find the application Id, you can e.g. open the applications table in the SER database

Modifying a StreamServer service

Prerequisites

A StreamServer service with a deployed export package must exist.

Command

```
ss_scm.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt" -servicename
MYSTRSSERVICE -action newarg -argname "-wd" -argvalue
"C:\Management Gateway\1.0\root\Applications\myapp"
```

Starting a StreamServer service

Prerequisites

A StreamServer service with a deployed export package must exist.

Command

```
ss_scm.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt" -servicename
MYSTRSSERVICE -action start
```

Checking if a StreamServer service is running

Prerequisites

A StreamServer service with a deployed export package must exist.

Command

```
ss_scm.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt" -servicename
MYSTRSSERVICE -action isrunning
```

Stopping a StreamServer service

Prerequisites

A StreamServer service with a deployed export package must exist.

Command

```
ss_scm.exe -user USERID -pass PASSWD -cert "C:\Program
Files\StreamServe\Platform\Core\1.4\bin\security\certificatestore
\trusted\authorities\streamserve.ca.crt" -servicename
MYSTRSSERVICE -action stop
```

Creating a web portal and assign it to a domain

1 Create an XML file, for example `MyPortal.xml`:

```
<?xml version="1.0 encoding="utf-8" ?>
<ResourceConfig>
    <!--PortalRoot is used by Management Gateway and Control Center
    when updating domain information for portal-->
    <PortalRoot>C:\Program Files\Apache Software Foundation\Tomcat
    5.5\webapps</PortalRoot>
</ResourceConfig>
```

or for unmanaged (manual deploy to Java application server)

```
<?xml version="1.0 encoding="utf-8" ?>
<ResourceConfig>
    <!--Used by Control Center for display. Not mandatory-->
    <ManualDeployPath>C:\Management
    Gateway\1.0\root\etc\unmanagedportals</ManualDeployPath>
</ResourceConfig>
```

2 Create a portal resource with the following command:

```
ss_territory.exe -action create_resource -resource_name
MyPortal -filename MyPortal.xml -resource_version <version>
-content_type application/x-streamserve.com-webportal
-strs_type application/x-streamserve.com-streamstudio -nodeid
<node_id>
```

3 Create a relation between the portal resource and an application domain with the following command:

```
ss_territory.exe -action assign_resource_to_territory_relation
-id 80652796-7020-4740-9E91-8F3A6764CC8D -idto <appdomain_id>
```

Creating an Archiver node

To create an Archiver node in the application domain, you perform the steps as in *Creating a web portal and assign it to a domain* on page 26 but the XML file contains a security profile holding a standard connection profile found in `territory.xml`.

Note: The `-content_type` argument is in this case `application/x-streamserve.com-archive` and the `-strs_type` argument is `application/x-streamserve.com-securityprofiles`.

Creating an Archiver application

The XML file you need for creating an Archiver application (as the `MyPortal.xml` in *Creating a web portal and assign it to a domain* on page 26) can be obtained by creating and configuring an Archiver application in Control Center and then retrieve the file from the working directory.

Then create an application with the ss_territory and ss_scm utilities (the `-apptype` argument is set to `STRSCI`) and create a resource and a relation as in *Creating a web portal and assign it to a domain* on page 26.

Note: The `-content_type` argument is in this case `application/x-streamserve.com-configuration` and the `-strs_type` argument is `application/x-streamserve.com-generic-app-configuration`

Troubleshooting

The utilities does not start

Solution

Check that there is an environment variable called `STRS_LOCATION` (in uppercase letters) that points to the Streamserver installation directory. Remember that in many UNIX dialects you must export the variable after you have defined it.

A StreamServer service does not start

Solution

The kernel XML file or the StreamServer .environment file may not be reference correctly in some cases. Also the application id may be required on some systems for the application to start.

To cover for this, you can add arguments to the service.

Run the following commands before you try to start the service again:

```
ss_scm -cert <path to cert> -action newarg -argname "-kernel" -
argvalue file://kernel_5_5_0.xml -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname "-env" -
argvalue "file://<path to the .environment file used by the
StreamServer>" -servicename <servicename>
```

```
ss_scm -cert <path to cert> -action newarg -argname "-
applicaitonid" -argvalue "<applicationid>" -servicename
<servicename>
```

Note: To find the application Id, you can e.g. open the applications table in the SER database

Prompt displayed for realm user name

(or e.g. `Error: System specific error code:146`)

This is probably because the management gateway connection is not working.

Solution

Check that:

- You have specified correct user name and password for the management gateway
- The management gateway is started.
- The `-ipport` argument specifies the correct port for your management gateway.

HTTP listener error displayed

```
Error: HTTPListener(scm): The file or a script interpreter is not
a regular file, execute permission is denied for the file or ascript
or ELF interpreter, the file system is mounted noexec or search
permission is denied on a component of the path prefix of filename
or the name of a script interpreter.
```

Solution

Make sure you specify the correct `-binpath` value when creating the service, i.e. the path to the start binary. For example `-binpath /var/streamserve/streamserve-5.5.0.GA.300/applicaitons/streamserver/start`

Note: The file must be included in the path, not just the path to the directory where the file is stored.