



Summary of Connecting to CAS and Accessing Data

Exploring the CAS Connection with CAS Actions

- Exploring a CAS Connection with Actions

Action	What it does
<code>session.listSessions</code>	Displays a list of the sessions on the server.
<code>session.sessionStatus;</code>	Displays the status of the current session.
<code>session.metrics / on=TRUE FALSE;</code>	Displays the metrics for each action after it executes.
<code>sessionProp.listSessOpts;</code>	Displays the session options and session values.
<code>sessionProp.getSessOpt / name="option to display";</code>	Displays the value of a session option.
<code>sessionProp.setSessOpt / timeout=seconds, metrics=TRUE FALSE <, ...>;</code>	Sets session option(s).
<code>builtins.serverStatus;</code>	Shows the status of the server.
<code>builtins.getLicensedProductInfo;</code>	Shows the information for licensed SAS products.
<code>builtins.getLicenseInfo;</code>	Shows the license information for a SAS product.
<code>builtins.actionSetInfo / all=TRUE FALSE;</code>	Shows the build information from loaded action sets.
<code>builtins.help / actionSet="action-set", action="action";</code>	Shows the parameters for an action or lists all available actions.

Exploring and Accessing Data Sources

- Caslibs

- A caslib is an in-memory space to hold tables, access control lists, and data source information. All data is available to CAS through caslibs and all operations in CAS that use data are performed with a caslib in place.
- A caslib can have session scope or global scope.
 - Session-scope caslibs are accessible only from the session that adds the caslib. This enables server-side data access to a programmer and does not interfere with other sessions. Use a session-scope caslibs when you need to access but not share tables.
 - Global-scope caslibs can be accessible to any session on the server. Depending on access controls, users can share access to in-memory tables. Personal caslibs and pre-defined caslibs are global caslibs. Global-scope caslibs are useful when you want other users to have access to the table, subject to access controls.
- Exploring Caslibs with Actions

Action	What it does
<code>table.caslibInfo;</code>	Shows all available caslibs and caslib information.
<code>table.fileInfo / caslib="caslib-name";</code>	Lists the available data source files in a caslib.
<code>table.tableInfo / caslib="caslib-name";</code>	Shows all available in-memory CAS tables in a caslib.
<code>table.addCaslib / name="caslib-name", path="folder-path", <, additional parameters>;</code>	Displays the value of a session option.

• LIBNAME Statement

- Create a SAS Compute server library reference name, or libref, that points to a caslib.

```
LIBNAME libref CAS <SESSREF="cas-session-name" CASLIB="caslib-name">;
```

• Managing In-Memory Tables

- Server-side files are files that are associated with a caslib.
- Client-side are files that are not associated with a caslib. They can be either:
 - Path-based files that are available to the CAS server, but are not associated with a caslib.
 - SAS data sets in SAS libraries that are available only to the compute server. These files need the CASUTIL procedure to load into CAS.
- Load a **server-side** file into memory:

```
table.loadTable /
  path="file-name.ext", caslib="caslib-name",
  casOut={name="table-name",
    caslib="caslib-name",
    replace= TRUE |FALSE...},
  importOptions={fileType="TYPE", ...},
  <additional parameters>;
```

- Load a **client-side path-based** file into memory:

```
table.upload /
  path="file-path",
  casOut={name="table-name"
    caslib="caslib-name" ...};
```

- Load a **client-side SAS data set** from a SAS library into memory:

```
proc casutil;;
  load data=SAS-data-set
    <casout="table-name">
    <outcaslib="caslib">
    <promote> <replace> <option(s)>;
quit;
```

- **Saving and Dropping CAS Tables**

- Save a CAS in-memory table to a caslib's data source:

```
table.save /
  table={caslib="caslib-name",
    name="table-name"},
  caslib="target-caslib", name="file-name.ext";
```

- Drop a CAS in-memory table:

```
table.dropTable /
  caslib="caslib-name",
  name="table-name"
  <quiet=TRUE|FALSE>
  ...;
```

- **CAS Table Scope**

- CAS tables have either session scope or global scope.
- Considerations for Session-scope tables
 - Best used for general purpose programming.
 - Typically provides better performance than global-scope tables because concurrency locks are not used.
- Considerations for Global-scope tables

- Best used for tables that are accessed by a large number of users, especially the visual interfaces.
- A global-scope table cannot be replaced. You must drop it and load the replacement data.

- Promoting Tables

- Promote a table when loading into memory:

```
table.loadTable /...  
casOut={...promote=TRUE};
```

```
table.upload /...  
casOut={...promote=TRUE};
```

- Promote an existing session scope table to global scope:

```
table.promote /  
name="table-name",  
caslib="caslib-name",  
<target="output-table-name">,  
<targetLib="output-casib">;
```