WHEN YOU NEED TO BE SURE

Internodal Communications Task Reference

Version 2

June 9, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	7/28/2021	Initial Publication
2	6/9/2022	Updated all hypertext linked cross-references to cyflex.com usage help descriptions

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion-buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1* Overview on page 1.

  The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1 Overview

CyFlex offers several applications that support rapid transfers of data from one system to another. Primary use of these applications is to share the value of a set of variables between systems. The applications support sharing of a centralized collection of facility-wide parameters such as barometer, combustion air duct humidity, fuel supply characteristics, natural gas composition, etc.

2 Internodal Communications Applications

2.1 connsrvr

`connsrvr` is TCP/IP based data transfer mechanism that is CyFlex version-independent. It is supported on Linux, MS-Windows, and QNX.

The application requires the following:

- A specification file, typically named `connsrvr_specs.`<*name*> to identify supported external nodes (IP addresses):
  - SERVERS: a list of nodes that provide a service to this node through `snode_link`, `push_link`, `getAstStat`, and other applications described later in this section.
  - DYNAMIC_SERVERS: a list of domains allowed to request data from this node

Refer to *Connection Server Setup* and cyflex.com usage help for `connsrvr` for related details.

2.1.1 connsrvr Specification File Example

```
# Connection Server Specfile
VERSION
1
# List remote nodes that provide a service to this node through
# the connection server.
# Examples of the server are sm_server, small_sm and  indicom.
# Examples of the applications requesting this service are
# node_link, snode_link, and getAstStat.
# HOST NAMES or IP ADDRESS
SERVERS
# Host           Remote
# NAME           Server Port
ctcnode45        -
ctcnode206       -
ctcnode230       -
# Windows Indicom Server
10.2.1.100       -
$
# List up to 5 ip address masks that the connection_server will accept
# ad-hoc requests from remote connection servers. These are remote
# connection servers that need a service such as sm_server or indicom.
#
# DYNAMIC_SERVER RANGE
DYNAMIC_SERVERS
143.222.0.0
# Windows Indicom Dynamic_Server
10.2.1.0
$
SERVER_TIMEOUT
300
$
```

2.2 csdump

Use `csdump` as a verification utility for `connsrvr`. This utility takes a snapshot of current communication status. Enter: `csdump | less`.

The output shows which nodes are communicating and whether there are networking errors.

Refer to *Connection Server Setup* and cyflex.com usage help for `csdump` for related details.

2.2.1 Sample csdump Output

csdump.1

```
Local Attached Services:
sm_serv_MAR09    pid(    22780) ipaddr 0.0.0.0
sm_serv_FEB06    pid(    22778) ipaddr 0.0.0.0
LV_inter         pid(    22779) ipaddr 0.0.0.0

sm_serv_MAR09  -  push_server
sm_serv_FEB06  -  small_sm
LV_inter_pid   -  web_sm_server
```

csdump.2

```
Remote Attached Services:
sm_serv_MAR09   pid(   23642)     ipaddr    XXX.XXX.XXX.171
sm_serv_FEB06   pid(   23640)     ipaddr    XXX.XXX.XXX.171
indicom         pid(    4196)     ipaddr    XXX.XXX.XXX.100
sm_serv_FEB06   pid(   18244)     ipaddr    XXX.XXX.XXX.86
LV_inter        pid(   18276)     ipaddr    XXX.XXX.XXX.86
sm_serv_MAR09   pid(    3830)     ipaddr    XXX.XXX.XXX.65
sm_serv_FEB06   pid(    3829)     ipaddr    XXX.XXX.XXX.65
sm_serv_FEB06   pid(   32589)     ipaddr    XXX.XXX.XXX.41
sm_serv_FEB06   pid(    3012)     ipaddr    XXX.XXX.XXX.165
sm_serv_FEB06   pid(    3618)     ipaddr    XXX.XXX.XXX.36
sm_serv_FEB06   pid(   15741)     ipaddr    XXX.XXX.XXX.39
sm_serv_FEB06   pid(    3306)     ipaddr    XXX.XXX.XXX.107
LV_inter        pid(    3308)     ipaddr    XXX.XXX.XXX.107
sm_serv_FEB06   pid(    3790)     ipaddr    XXX.XXX.XXX.98
sm_serv_FEB06   pid(    2977)     ipaddr    XXX.XXX.XXX.95
sm_serv_MAR09   pid(    2978)     ipaddr    XXX.XXX.XXX.95
sm_serv_FEB06   pid(    6078)     ipaddr    XXX.XXX.XXX.36
```

csdump.3

Connected Remote Servers:

Hostname/IP	Alive Time	Listen Taskid	SendTo Taskid	Msg Count	Packet Count	Error Count	Last Time	Send Taskid	Requestor Taskid	Msg Count	Packet Count	Error Count	Last Time
XXX.XXX.XXX.171	09:24:40	27042	0	0	0	0	19:00:00	27043	0	0	0	0	19:00:00
XXX.XXX.XXX.86	09:24:40	23093	0	0	0	0	19:00:00	23094	0	0	0	0	19:00:00
XXX.XXX.XXX.153	09:24:40	23011	0	0	0	0	19:00:00	23012	0	0	0	0	19:00:00
XXX.XXX.XXX.65	09:25:45	20612	22778	907	43	0	09:26:32	20613	22778	878	90	0	09:26:32
XXX.XXX.XXX.44	09:26:02	20607	0	0	0	0	19:00:00	20608	0	0	0	0	19:00:00
XXX.XXX.XXX.41	09:26:31	20599	22779	612	807	28	08:51:41	20600	22779	584	13092	0	08:51:41
XXX.XXX.XXX.42	09:26:31	20597	0	0	0	0	19:00:00	20598	0	0	0	0	19:00:00
XXX.XXX.XXX.165	09:26:02	20595	0	0	0	0	19:00:00	20596	0	0	0	0	19:00:00
XXX.XXX.XXX.39	09:22:21	20449	22778	17	17	0	07:29:14	20450	22778	17	17	0	07:29:14
XXX.XXX.XXX.252	09:22:21	20447	0	0	0	0	19:00:00	20448	0	0	0	0	19:00:00
XXX.XXX.XXX.155	09:24:40	20445	0	0	0	0	19:00:00	20446	0	0	0	0	19:00:00
XXX.XXX.XXX.36	09:26:31	20443	0	0	0	0	19:00:00	20444	0	0	0	0	19:00:00
XXX.XXX.XXX.208	09:22:21	20441	0	0	0	0	19:00:00	20442	0	0	0	0	19:00:00
XXX.XXX.XXX.107	09:22:21	20439	22778	155259	155259	60	09:26:36	20440	22778	155199	155199	0	09:26:36
XXX.XXX.XXX.98	09:22:21	20437	22778	155259	155259	60	09:26:38	20438	22778	155199	155199	0	09:26:38
XXX.XXX.XXX.100	09:26:40	20342	6905	197679	197679	0	15:37:25	20357	6905	197714	197714	0	15:37:25
node230	09:24:40	20343	0	0	0	0	19:00:00	27009	0	0	0	0	19:00:00
node206	09:14:38	20344	0	0	0	0	19:00:00	27010	0	0	0	0	19:00:00
node45	09:24:41	20345	22807	25865	25865	0	09:26:05	27011	22807	25865	25865	0	09:26:05

2.3 small_sm

The `small_sm` application maintains awareness of the organization of shared memory on the local node. It responds to a version-independent message from remote nodes that contain:

- Value
- Units
- Display format
- Last update time

`small_sm` communicates with multiple remote nodes and supports only individual members of data types which contain multiple values (statistical, composition, etc.).

Refer to *Connection Server Setup* and cyflex.com usage help for `small_sm` for related details.

2.4 small_smN

The `small_smN` application is identical to `small_sm` except that works exclusively with only one remote node to avoid contention with other nodes.

Launch the application with an argument to identify the remote service it supports as in the following example:

```
small_smN em_cart &
```

The remote node must run `node_linkN` using the same identifier as its `extender` optional argument as in the following example:

```
node_linkN extender=em_cart &
```

Refer to cyflex.com usage help for `small_smN` and for `node_linkN` for related details.

2.5 snode_link

Use the `snode_link` application to fetch or push the value of variables between remote nodes. The application supports multiple nodes and multiple instances. It is recommended to isolate high data rates, for example, greater than 1[hz].

Refer to *Connection Server Setup* and cyflex.com usage help for `snode_link` for related details.

2.6 node_linkN

The `node_linkN` application is identical to `snode_link` except that it communicates with a remote instance of `small_smN`. The application supports multiple nodes and multiple instances. It is recommended to isolate high data rates, for example, greater than 1[hz].

Refer to *Connection Server Setup* and cyflex.com usage help for `node_linkN` for related details.

2.7 push_link

The `push_link` application is similar to `snode_link`, but only sets variable values rather than retrieving them.

Refer to cyflex.com usage help for `push_link` for related details.

2.8 push_server

The `push_server` application enables transmissions from remote `push_link` instances. It is similar to the `small_sm` application, but only sets variable values rather than retrieving them.

Refer to cyflex.com usage help for `push_server` for related details.

2.9 getAstStat13

Use the `getAstStat13` application to perform a “node link” type of operation to fetch certain variables from all CyFlex test cells. Typically, this application is run on the central node. It uses the `/data/cell_map` file to determine the list of nodes from where to fetch the variables. The variables are automatically created and added to the “save list”.

A specification file is not required.

The fetched variables are:

hour

`test_hrs`	`total_work`	`abs_work`
`mot_work`	`eng_hrs`	`total_fuel`
`last_on_time`	`last_off_time`	`util_7day`
`go_count`	`down_code`	`run`
`eng_Model`	`sn`	`test_type`
`engineer`	`technician`	`Test_ID`
`prog_proj`		

Refer to cyflex.com usage help for `getAstStat13` for related details.

2.10 getAstStatGen

The `getAstStatGen` application is identical to the `getAstStat13` application except that the variables are configurable by the `/specs/fetch_remote` file. It uses the `/data/cell_map` file to determine the list of nodes from where to fetch the variables listed above for `getAstStat13`. The variables are automatically created and added to the “save list”.

The specification file may include up to 20 variables in the following sample format:

```
#label    data_type
My_xzy    REAL_TYPE
My_int    INT_TYPE
My_logi   LOGI_TYPE
My_c      STRING_TYPE
```