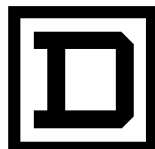


CRISP connect™ **Server** *for @aGlance/IT™*

User's Guide

§ CRISP Software Products



SQUARE D
GRUPE SCHNEIDER

**CRISP connect™ Server for @aGlance/IT™
User's Guide**

Document number: 500 070 - 001, Rev. 1

Document History

Revision	Date	Pages affected/Description of change
1	10/93	Initial release per ECN 4315

Software Version 1.0

This information furnished by Square D Company is believed to be accurate and reliable. However, Square D Company neither assumes responsibility for its use nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Square D Company. This information is subject to change without notice.

Copyright 1993 by
Square D Company
5160 Paul G. Blazer Memorial Parkway
Dublin, Ohio 43017
USA

WARNING: Any unauthorized sale, modification or duplication of this material may be an infringement of copyright.

CRISP® is a registered trademark of Square D Company.

CRISP®/32 is a registered trademark of Square D Company.

CRISPconnect™ is a trademark of Square D Company. The CRISPconnect family of products includes IDI, WORF, DATAGATE, an @aGlance/IT Server, and a NetDDE Server.

@aGlance/IT™ is a trademark of Intuitive Technology Corporation.

The following are trademarks of Digital Equipment Corporation: VMS, DEC, VAX, MicroVAX, DECnet, and Pathworks.

Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation.

Introduction	General1 Intended Audience.....2
Installing CRISPconnect Server	Requirements3 CRISPconnect Server Installation5 Register the License.....5 Configuring CRISPconnect Server.....5 Installation of Non-CRISP Software6 DEC ACA Services6 @aGlance/IT.....6 Configure ACA Services for @aGlance6
Accessing CRISP Data	General9 CRISPconnect Server Service Names.....9 Tag and Attribute.....9 @aGlance Service Supported 10 Get Tagnames 10 Get Attributes..... 11 Get List11 Get Table 11 Put List11 Put Table..... 11 Monitor..... 11 Get History 12

Notes:

General

The CRISPconnect™ *Server for @aGlance/IT™* for CRISP computer-based systems enables popular desktop applications (such as spreadsheets, graphical packages, and statistical packages) to be integrated with CRISP process-management systems via an open client/server architecture. The client/server architecture is based upon @aGlance/IT from Intuitive Technology Corporation, which is a standard interface specifically designed to facilitate the integration of desktop tools with manufacturing control systems across multi-vendor platforms. In addition to desktop applications, any "client" application (such as a Plant Floor Management system) that supports @aGlance/IT can be integrated with CRISP systems via this *CRISPconnect Server*.

This *CRISPconnect Server* performs the role of an @aGlance/IT "server". It functions as a data manager to access CRISP data on behalf of requests from "client" @aGlance/IT applications running on other nodes and other platforms. The information exchange between CRISP and these client applications provides interactive access to manufacturing data in CRISP systems for data display, analysis and reporting. The information exchange is performed transparently for both the application developer and the end user.

This *CRISPconnect Server* offers the following @aGlance/IT features:

- Allows users to choose the desktop application they want, run it on their own hardware, and access process data on a CRISP system.
- Supports many popular desktop platforms and applications.
- Allows desktop and other client applications to exchange data with one or more servers distributed over a network, where this *CRISPconnect Server* may co-exist with other @aGlance/IT servers.
- Provides fast and efficient data transmission.
- Provides interactive access to process data.
- Allows the sharing and distribution of process data.
- Manages incompatibilities in a multi-vendor environment.

This *CRISPconnect Server* conforms to the @aGlance/IT standard interface. It utilizes the process data model and the service model provided by @aGlance/IT to identify and access CRISP data. It supports access to both real-time data and historical data in CRISP systems.

The data model used to reference process data consists of a *tagname* and optional *attributes*. A *tagname* can identify either a block of data or a single element of data. In the case where a *tagname* identifies a block of data, each item within the block is identified by an *attribute* name.

The service model used to access CRISP data provides the kinds of requests an @aGlance/IT client application invokes to exchange manufacturing data.

Intended Audience

The User's Guide for the *CRISPconnect Server for @aGlance/IT* is intended for use as a reference document for personnel who are responsible for the system administration of the CRISP systems purchased from Square D Company. It also is intended for use as a reference document for application developers who need to have an understanding of the functions supported by this *CRISPconnect Server*.

This user's guide does not address how to use @aGlance/IT functions in "client" applications to access CRISP data. Such information is described in the product documentation provided by each vendor whose product is an @aGlance/IT "client" application.

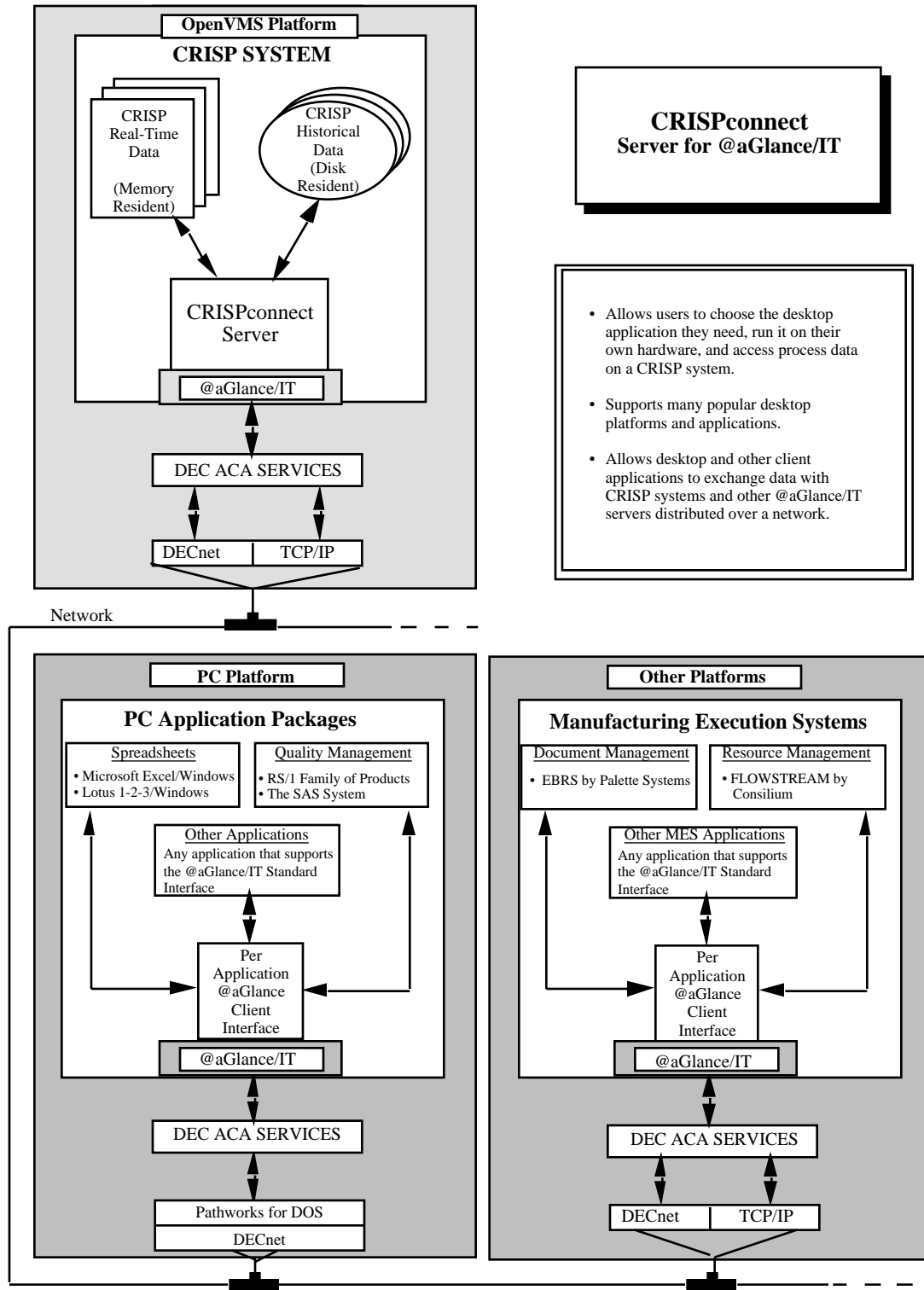
Requirements

This *CRISPconnect Server* requires installation of the following software (reference diagram on the following page entitled *CRISPconnect Server for @aGlance/IT*).

- VMS Operating System V5.5, or later
- DECnet for VMS V5.5, or later
Optionally, DEC TCP/IP V2.0
- DEC ACA Services for VMS V2.1 (or later) Runtime Option. (DEC ACA Services is included with NAS 250, NAS 300, and NAS 400 software packages.)
- @aGlance/IT for VMS V1.0 (or later) Server Runtime Option
- CRISP[®]/32 System Software, V3.0

The client platform that will be accessing this CRISPconnect Server requires installation of the following software to support @aGlance/IT:

- Software that supports DECnet, Digital's TCP/IP, or Novell's TCP/IP. On a PC, DEC's PATHWORKS for DOS V4.1 can be used.
- DEC ACA Services for the Client platform.
- @aGlance/IT interface for the Client Product.



**CRISPconnect
Server for @aGlance/IT**

- Allows users to choose the desktop application they need, run it on their own hardware, and access process data on a CRISP system.
- Supports many popular desktop platforms and applications.
- Allows desktop and other client applications to exchange data with CRISP systems and other @aGlance/IT servers distributed over a network.

CRISPconnect Server for @aGlance/IT

CRISPconnect Server Installation

To install this *CRISPconnect Server*, its license must be loaded and its software must be configured as described in the following two sections.

Register the License

Register the *CRISPconnect Server* license through the VMS License Management Facility (LMF) in accordance with the license agreement signed for your site. The license registration information you need is contained in the Product Authorization Key (PAK) that is shipped with this *CRISPconnect Server*. The PAK is a paper certificate that contains information about your license to run a particular piece of software.

You must register and load your *CRISPconnect Server* license before the server can run in a production environment. If the license is not loaded, this *CRISPconnect Server* will run for one hour in demo mode and will then terminate.

To register a license under VMS, log into the system manager's account and issue the following DCL command:

```
$ @SYS$UPDATE:VMSLICENSE
```

As the VMSLICENSE procedure prompts you for information, respond with data from your Product Authorization Key (PAK). For complete information on how to use the license utility, see the *VMS License Management Utility Manual* in the Base VMS Documentation Set.

Configuring CRISPconnect Server

This software is configured by logging into the CRISP account and executing the CRISP_CONFIG command procedure to define the configuration of the VAX and CRISP/32 system hardware and software. The user typically executes the CRISP_CONFIG command after a new CRISP/32 system has been installed or after installing a new release of CRISP/32 on an existing system.

When executed, a portion of the CRISP_CONFIG command procedure prompts the user for the configuration of this *CRISPconnect Server*. For additional information on the CRISP_CONFIG command procedure, reference the *CRISP/32 Utilities Reference Manual*.

When the CRISP_CONFIG command procedure is complete, the *CRISPconnect Server* processes will automatically start when CRISP is started. However, before the *CRISPconnect Server* processes will successfully run, the Non-CRISP software identified in the next section must be installed.

Installation of Non-CRISP Software

As noted above, in addition to VMS and DECnet which most VAX systems already have installed, DEC ACA Services and @aGlance/IT must also be installed to support this *CRISPconnect Server*.

DEC ACA Services

To install DEC ACA Services, follow the instructions in the *DEC ACA Services Installation Guide* which guide the user through the following steps:

1. Register the DEC ACA Services License PAK using @SYSS\$UPDATE:VMSLICENSE.
2. Install the DEC ACA Services software using @SYSS\$UPDATE:VMSINSTAL.

@aGlance/IT

To install @aGlance/IT, follow the instructions in the *@aGlance/IT Installation Guide* which guide the user through the following steps:

1. Register the @aGlance/IT License PAK using @SYSS\$UPDATE:VMSLICENSE.
2. Install the @aGlance/IT software using @SYSS\$UPDATE:VMSINSTAL.

Configure ACA Services for @aGlance

DEC ACA Services must be configured for @aGlance/IT and this *CRISPconnect Server*. This configuration sets up the security for @aGlance/IT to grant access to the server for incoming node and username requests. It also sets up the security for this *CRISPconnect Server* to grant access for incoming read/write requests that have passed the @aGlance/IT security check.

The following steps give an overview of the items to configure and how to perform the configuration (reference the *@aGlance/IT Installation Guide* for more information):

1. Log into the system manager's account.
2. Invoke the @aGlance/IT Administration Tool as follows:

Type: MCR AAG_ADMIN

A menu appears that allows adds, deletes, and viewing to be performed on the ACA Services database for @aGlance configuration.

3. Select the menu option to add the node and platform type of all nodes that will be accessing this server node.
4. Select the menu option to add the proxies of all nodes and usernames that are to be granted access to this server and identify the account their access is to run under.

Configure ACA Services for @aGlance (cont)

5. Select the menu option to add permissions of all nodes and usernames that are to be granted read and/or write access to this server.

Read permission is granted for a given node and username by this *CRISPconnect Server* if they have been configured to have "CRISPAAG_READ" permission.

Write permission is granted for a given node and username by this *CRISPconnect Server* if they have been configured to have "CRISPAAG_WRITE" permission.

6. Select the menu option to commit the updated configuration to the ACA Services database.
7. Exit the @aGlance/IT Administration Tool.
8. Load the new configuration into the ACAS Services run-time database by invoking the ACA Services Manager as follows (reference the manual entitled *DEC ACA Services - System Manager Guide for VMS Systems* for more information):

Type: APPLICATION/CONTROL SET CTRL_SERVER/NEW_CONTEXT

9. In the event that the ACAS Control Server "ACAS\$CONTROL" needs to be shutdown, use the following command:

Type: @SYS\$MANAGER:ACAS\$SHUTDOWN

10. In the event that the ACAS Control Server "ACAS\$CONTROL" needs to be started, use the following command:

Type: @SYS\$STARTUP:ACAS\$STARTUP

11. Log out of the system manager's account.
12. Log into the CRISP account, start CRISP, and the *CRISPconnect Server* processes will automatically be started up and CRISP data should be accessible to all @aGlance/IT "clients" whose nodes and usernames were configured above.

Notes:

General

This section describes the functions supported by this *CRISPconnect Server* to provide the application developer with an understanding of how to access CRISP process data from an @aGlance/IT "client" application.

CRISPconnect Server Service Names

This *CRISPconnect Server* must be present locally on each CRISP system that is to be accessed by @aGlance/IT client applications. At startup, a *CRISPconnect Server* registers its service name with ACA Services, which makes the name available to all @aGlance/IT client applications.

Client applications request access to a specific CRISP system by specifying the unique service name of a *CRISPconnect Server* on that CRISP system. Each CRISP system has two *CRISPconnect Server* services whose service names are as follows:

1. The service name of the Server that provides access to real-time data is: "CSRAAG_NodeName", where 'NodeName' is the DECnet node name of the CRISP system.
2. The service name of the Server that provides access to historical data is: "CSHAAG_NodeName", where 'NodeName' is the DECnet node name of the CRISP system.

Tag and Attribute Names

Each client request to access CRISP data includes a list of the CRISP *tagnames* and their optional *attributes* that are to be accessed as described below:

1. A CRISP *tagname* is used to reference data in a CRISP system and is defined as "Database:Variable(Subscript)", where "(Subscript)" is optional. The Subscript is used to reference an array element of a Variable and may be a numeric constant or another Variable.
2. A CRISP *attribute* is used to reference a specific data element of a *tagname*. Multiple *attributes* of a *tagname* can be accessed while using only one reference to the *tagname*. This allows attributes that correspond to multiple elements of a Variable (like array element indexes) to be used to access a block of data for a given *tagname*.

The attributes supported by this *CRISPconnect Server* are listed and described below.

Self:	denotes a simple tagname, which is the default if an attribute is not specified.
Reset:	specifies the reset element of a TIMER or COUNTER variable.
Tickdown:	specifies the tickdown element of a TIMER variable.
Count:	specifies the count element of a COUNTER variable.

Tag and Attribute Names (cont)

(Sizeof): specifies the size of an array variable on a read; invalid on a write.

(Subscript): specifies an array element of a variable; may be a numeric constant or another variable.

3. This *CRISPconnect Server* supports access to tagnames in a CRISP system that are of the following CRISP data types.

STRING, LOGICAL, NUMERIC, LONG, FLOAT, COUNTER, TIMER

@aGlance Services Supported

This *CRISPconnect Server* supports all of the standard @aGlance/IT V1.0 services for a "Client" to access CRISP process data. Each service is briefly described below.

This user's guide does not address how to use @aGlance/IT functions in "client" applications to access CRISP data. Such information is described in the product documentation provided by each vendor whose product is an @aGlance/IT "client" application.

Get Tagnames

The GET TAGNAMES service returns a list of known CRISP real-time databases and variables that exist on a given CRISP system. This service supports the capability to use wildcard characters (* and ?) as a match-string filter on the tagnames that are to be returned.

If the filter does not include a database name and colon, this service returns a list of only the CRISP databases matching the filter. If the filter includes a database name and colon, this service returns a list of the variables in the specified database matching the filter.

Filter Examples: database:variable(subscript)
 database:variable
 database:*
 *

The GET TAGNAMES service also provides the capability to obtain a list of CRISP historical data files. By prefixing the match-string filter with "HIST>", a list of the matching CRISP Historian Point Files will be returned. The match-string filter can be specified using two different methods, where the method is determined by the syntax of the match-string filter. The two syntax methods are as follows:

1. The "DEFAULT" method uses the CRISP Historian default directory logical (e.g., CRISP\$HIST_ROOT) to search for matching point file names. This method uses the same tag naming convention as that used to access real-time data plus it allows a "Client" to access historical data on other nodes.

Get Tagnames (cont)

Filter Examples: HIST>node::database:variable(subscript)
HIST>database:variable(*)
HIST>database:variable
HIST>node::*
HIST>*

2. The "USER-SPECIFIED" method uses the search file specification directly as specified by the "Client". This method allows users to search-for-and-access Historian Point Files that reside in directories other than in the CRISP\$HIST_ROOT directory path. In all cases, the file extension appended to the search file specification is ".HP*".

Filter Examples:

```
HIST>node::device:[directory]variable(subscript)
HIST>node::device:[directory]variable(*)
HIST>node::logical_directory:variable
HIST>node::logical_directory:*
HIST>logical_directory:variable
HIST>logical_directory:*
```

Get Attributes

The GET ATTRIBUTES service returns a list of all attributes for a specified list of tagnames. This service supports the capability to use wildcard characters (* and ?) as a filter on the attributes that are to be returned.

The only attribute returned on a tagname that is an Historian Point File name is "SELF".

Get List

The GET LIST service returns the values for a specified list of tagname and attribute pairs, where attributes do not have to be specified. If attributes are not specified, the attributes default to SELF.

Get Table

The GET TABLE service returns the values for a specified table of tagnames and attributes. For m tagnames and n attributes, $m \times n$ values are returned.

Put List

The PUT LIST service writes the values for a specified list of tagname and attribute pairs, where attributes do not have to be specified. If attributes are not specified, the attributes default to SELF.

Put Table

The PUT TABLE service writes the values for a specified table of tagnames and attributes. For m tagnames and n attributes, $m \times n$ values are written.

Monitor

The MONITOR service monitors a specified list of datapoints which consists of tagname and attribute pairs, where attributes do not have to be specified. If attributes are not specified, the attributes default to SELF. At the initial request, the values of all the datapoints are returned. Thereafter the datapoints are monitored according to the MONITOR request.

Monitor (cont)

Each MONITOR request can optionally specify the time interval at which the datapoints are to be monitored in increments of one second. If the time interval is not specified in the MONITOR request, the monitor time interval defaults to three seconds. The time interval is specified in VMS format delta time; for example, one second is specified as follows: "0 00:00:01".

Each MONITOR request can optionally specify the type of monitoring to be performed, which is either 1) "Data-By-Exception" or 2) "Poll-By-Time". If not specified, the monitoring type defaults to "Data-By-Exception". The "Data-By-Exception" monitoring type returns the values of all datapoints in the request set when any datapoint value changes. The "Poll-By-Time" monitoring type returns the values of all datapoints in the request set at the expiration of every monitor time interval.

Get History

The GET HISTORY service returns the historical values for a specified list of tagname and attribute pairs, where attributes do not have to be specified. If attributes are not specified, the attributes default to SELF.

This service transforms the specified tagnames and their optional attributes into the CRISP Historian Point File naming format. The tagnames are specified by the "Client" using one of the two syntax methods described in section **Get Tagnames**; however, the "HIST>" prefix does not have to be specified.

The historical time range is specified as a start time, an interval time, and a number of intervals. For each time interval, a value for each of the tagnames is returned. For m intervals and n tagnames, m x n historical values are returned.

The start time can be specified in one of three ways:

- 1) VMS format absolute time; specified as "DD-MMM-YYYY HH:MM:SS".
- 2) Find "First" historical data point; specified as "FIRST".
- 3) Find "Last" historical data point; specified as "LAST".

The interval must always be specified in VMS format delta time as "0 HH:MM:SS"; for example, three seconds is specified as "0 00:00:03".