

C•CURE 800/8000

Version 9.1

Revision E0

C•CURE 800/8000 ODBC Configuration Guide

A Guide for Obtaining Data from the
C•CURE Database via an ODBC Connection

Software House
70 Westview Street
Lexington, MA 02421-3108
<http://www.swhouse.com>
Fax: 781-466-9550 Phone: 781-466-6660

C•CURE® is a registered trademark of Sensormatic Electronics Corporation. All other brand and product names are the trademarks of their respective owners. Information furnished by Software House is believed to be accurate. However, no responsibility is assumed by Software House for its use, nor for any infringement of rights of third parties which may result from its use. Specifications are subject to improvement or change without notice.

This manual is proprietary information of Software House. Unauthorized reproduction of any portion of this manual is prohibited. The material in this manual is for informational purposes only. It is subject to change without notice. Software House assumes no responsibility for incorrect information this manual may contain.

CCURE 800/8000 Version: 9.1
Document Revision: E0

Copyright © 2004 - 2006 by Sensormatic Electronics Corporation

All rights reserved

Table of Contents

Chapter 1	1-1
Introduction	1-1
Overview	1-2
Supported ODBC Driver	1-2
Software Requirements	1-2
Chapter 2	2-1
Configuring ODBC Using the PROGRESS SQL92 Driver	2-1
Overview	2-2
Setting up Database Accounts	2-2
Configuring the C•CURE Server	2-5
Setting up the ODBC Client for the CF Database	2-5
Using Progress SQL Explorer	2-11
Granting ODBC Access Rights to Users	2-12
Using SQL Explorer to Execute SQL92 Statements	2-14
SQL Problems and Solutions	2-24
Contact Information	2-24
Setting up the ODBC Client for the Journal Database	2-25
Using ODBC with a Prior Journal	2-26
Starting and Stopping a Journal Server via a Batch File	2-27
Chapter 3	3-1
Database Schema	3-1
Overview of the Database Schema	3-2
How the CF (Configuration) Database is Organized	3-3
Table Layouts	3-3
Database Schema Conventions	3-4
Clearance Object	3-8
Door Object	3-12
Groups Object	3-16
Inputs Object	3-18
Outputs Object	3-21
Person Object	3-24
Reader Object	3-41
Time Spec Object	3-44
Time Zone Object	3-47
Unit Object	3-49
Journal Object	3-56
Reference Tables	3-59
Chapter 4	4-9
Accessing Journal Information	4-9
How the Journal Database is Organized	4-10
Looking up Journal Information	4-11
Finding the Current Journal Database	4-11
Compressed Integer Date/Time Format	4-12
Decoding the Compressed Integer Date/Time Value	4-12
Journal Table Structure and Message Codes	4-16

Chapter 1

Introduction

The *C•CURE 800/8000 ODBC Configuration Guide* describes how to access data from Version 9.1 of the C•CURE 800/8000 database tables using ODBC (*Open Database Connectivity*) enabled applications, such as Microsoft Access, Microsoft Visual Basic, Microsoft Excel, or Crystal Reports.

You can also import data from ODBC applications into C•CURE 800/8000 databases, using the *Automated Personnel Import*, *Manual Personnel Import*, or *CF Database Import/Export Tool* functions.

For information about these import features, refer to the *C•CURE 800/8000 Advanced User Guide* or to the C•CURE 800/8000 online help for the Administration application.

Note ODBC access to the C•CURE 800/8000 databases is restricted to read-only access. Only the 32-bit ODBC drivers are supported.

This guide is intended for use by experienced ODBC users using *C•CURE 800/8000 Version 9.1*.

In This Chapter

- Overview
- Supported ODBC Driver
- Software Requirements

Overview

C•CURE 800/8000 Version 9.1 uses the PROGRESS® Database Management System (DBMS) to store and retrieve information.

- **CF Database** – The *CF* database stores configuration information in tables, which are collections of logically related records that the software views as a unit.

For example, the C•CURE 800/8000 Personnel table is a collection of records containing information about personnel, that is, cardholders and users. The Personnel table contains one record for each cardholder and/or user entered in the system. Refer to Chapter 3 for table descriptions.

- **Journal Database** – The Journal database stores real-time information about card accesses and other system activity. Each journal is a separate database. For information about journals, date/time values, and naming conventions, refer to Chapter 4

ODBC is a standard database access method that allows you to use a client software application, such as Microsoft Visual Basic, Access, or Excel to access the C•CURE 800/8000 database tables to retrieve information. The *ODBC driver* is a software component that “translates” ODBC data queries from the client application, such as Microsoft Visual Basic, into commands that the database engine on the server understands.

Supported ODBC Driver

The **MERANT 3.60 32-Bit PROGRESS SQL92 V9.1D** driver is ODBC-compliant and supported for access to the C•CURE 800/8000 Progress database.

This driver is included with C•CURE 800/8000, and is automatically installed and licensed for use on any PC which has either the C•CURE 800/8000 client or server software installed. For information about the **ODBC** driver, refer to Chapter 2

Software Requirements

The following software is required:

- C•CURE 800/8000 client or server software Version 9.1, which includes the required Progress software.
- MERANT 3.60 32-Bit PROGRESS SQL92 V9.1D ODBC driver, which is installed with the C•CURE 800/8000 software.
- An ODBC compliant software package, such as Microsoft SQL Server, Microsoft Access, Microsoft Excel, Visual Basic, or Crystal Reports. A runtime version of Crystal Reports is installed with C•CURE 800/8000 software for use with pre-defined reports.

Configuring ODBC Using the PROGRESS SQL92 Driver

This chapter describes how to set up ODBC applications on both the C•CURE System server and C•CURE client machines for read-only access with the MERANT 3.60 32-Bit PROGRESS SQL92 V9.1D Driver.

To use the Automated Personnel Import feature, refer to the C•CURE 800/8000 online help for the Administration application.

In This Chapter

- Overview
- Setting up Database Accounts.
- Configuring the C•CURE Server
- Setting up the ODBC Client for the CF Database
- Using Progress SQL Explorer
- Granting ODBC Access Rights to Users
- Using SQL Explorer to Execute SQL92 Statements
- SQL Problems and Solutions
- Setting up the ODBC Client for the Journal Database
- Using ODBC with a Prior Journal
- Setting up the ODBC Client for the Journal Database
- Starting and Stopping a Journal Server via a Batch File

Overview

These instructions are for configuring the MERANT 3.60 32-Bit PROGRESS SQL92 V9.1D ODBC driver for read-only access.

Caution The instructions below require that C•CURE 800/8000 V 9.1 is installed on a dedicated system.

Setting up Database Accounts

Use the C•CURE 800/8000 Administration application to create user accounts to access the database.

- Database Administrator account – Create a Database Administrator (DBA) named SYSPROGRESS. Once created, the SYSPROGRESS user has *full database administration privileges* so this user account should be managed carefully.
- Additional accounts – Create additional database accounts with read-only privileges to connect to the ODBC application and perform such functions as executing SQL statements to query the database.

You can assign any names for the ODBC user accounts; for example, odbuser, fred_odbc, or tjones. However, these additional users have no ODBC access rights to the database until they are allocated read-only privileges with the SQL GRANT command.

To grant read only privileges, log into the SQL Explorer application with the SYSPROGRESS account. Refer to the following sections for detailed information.

Setting up the SYSPROGRESS Account

To set the SYSPROGRESS user name and password for the database administrator:

1. Open the **C•CURE 800/8000 Administration** application.
2. Add or edit a personnel record for the SYSPROGRESS account.

The first and last name fields are arbitrary; however, you must enter SYSPROGRESS or sysprogress (case insensitive) in the User Name field. You can create a system-level account by leaving the first name field blank and entering *SYSPROGRESS* in the last name field.

3. Select these checkboxes: *User*, *User Enabled*, *ODBC Enabled*, and *Breakthrough Enabled*
4. Enter the User Name (Login) as *SYSPROGRESS* or *sysprogress*.
5. Specify a User password and ODBC password.

The screenshot shows the 'Edit Personnel Record' window with the 'User' tab selected. The 'Last Name' field contains 'sysprogress' and the 'User Name' field contains 'sysprogress'. Both fields are circled in red. The 'User', 'User Enabled', 'ODBC Enabled', and 'Breakthrough Enabled' checkboxes are all checked. The 'Privilege Restrictions' section shows 'Admin Priv.' set to '\$Full Administration', 'Monitor Priv.' set to '\$Full Monitoring', and 'Personnel Viewr.' set to '\$System Person View'.

Figure 1: Setting up the SYSPROGRESS Account

Setting up an ODBC User Account

To set the ODBC user name and password:

1. Open the **C•CURE 800/8000 Administration** application.
2. Add or edit a personnel record for each user who will have read-only access to the ODBC database.
3. On the *User* tab, enter the name of the person with read-access to the ODBC database, such as *ODBCUSER*, *fred_odbc*, or any meaningful name.

NOTE: You can enter any name in the Personnel Record of a user with ODBC access; however, including "odbc" as part of the name makes it easier to identify the account function.

4. In the *User Information* area, enter any meaningful User Name (Login), for example, *odbcuser*.
5. Select these checkboxes: *User*, *User Enabled*, *ODBC Enabled*, *Breakthrough-Enabled*,
6. Specify a User password and ODBC password.

The screenshot shows the 'Add Personnel Record' window with the following details:

- Form Fields:** First Name, M.I., Last Name (containing 'odbcuser'), Card #, Person Type (dropdown menu), Person ID (containing '2051').
- User Information Section:** Includes checkboxes for 'User', 'User Enabled', 'ODBC Enabled', and 'Breakthrough Enabled', all of which are checked. It also features 'Set User Password' and 'Set ODBC Password' buttons.
- Privilege Restrictions Section:** Includes 'Admin Priv.' (set to '\$Full Administration'), 'Monitor Priv.' (set to '\$Full Monitoring'), and 'Personnel View' (set to '\$System Person View').
- Navigation:** A tabbed interface with 'User' selected. Other tabs include 'General', 'Clearances', 'Badging', 'Customer', 'Prev. Doors', and 'Custom Access'.

Figure 2: Setting up an ODBC User Account

Configuring the C•CURE Server

The C•CURE 800/8000 server must have C•CURE 800/8000 Version 9.1 installed. There are no special procedures to set up the C•CURE 800/8000 server for ODBC.

Setting up the ODBC Client for the CF Database

When you install C•CURE 800/8000 Version 9.1 on the ODBC client, the SQL92 driver is also installed.

This section describes how to define the ODBC data source on the client PC. The data source specifies the link from the client ODBC application to Progress databases on the server.

Before defining an ODBC Data Source, make sure that the C•CURE 800/8000 database is up and running.

Note Follow a similar procedure to connect to a journal database. For information about setting up and configuring Journal databases, refer to the section *Setting up the ODBC Client for the Journal Database* later in this chapter.

To add an ODBC data source

1. Select **Settings > Control Panel > Administrative Tools** from the Windows *Start* menu.

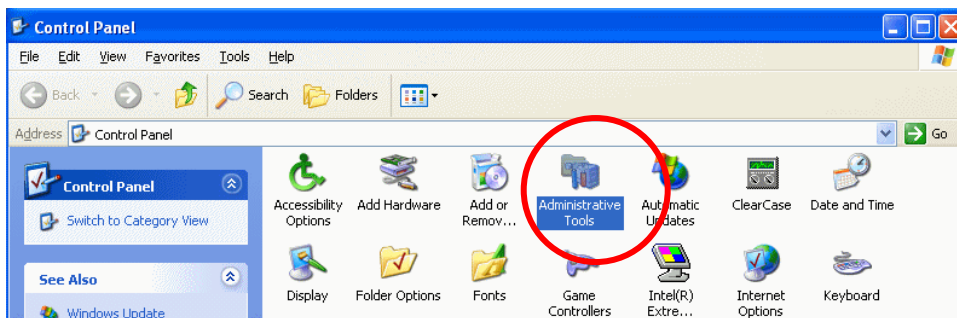


Figure 3: Selecting Administrative Tools from the Control Panel

2. Double-click the **Data Sources (ODBC)** icon in the Administrative Tools window.

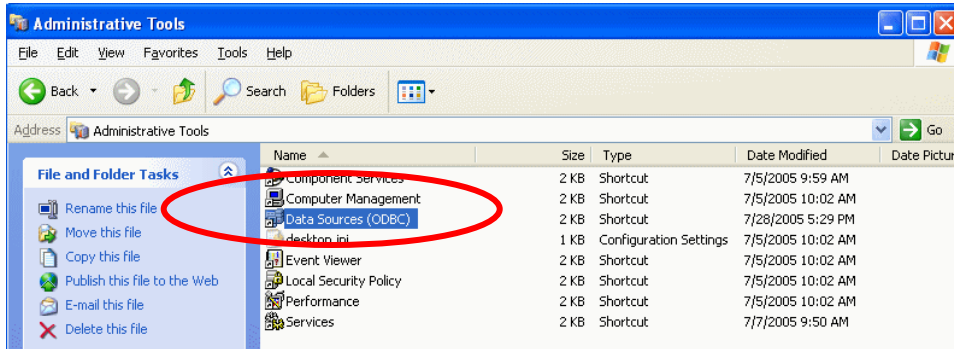


Figure 4: Selecting the Data sources (ODBC) Icon

3. The *ODBC Data Source Administrator* window appears:

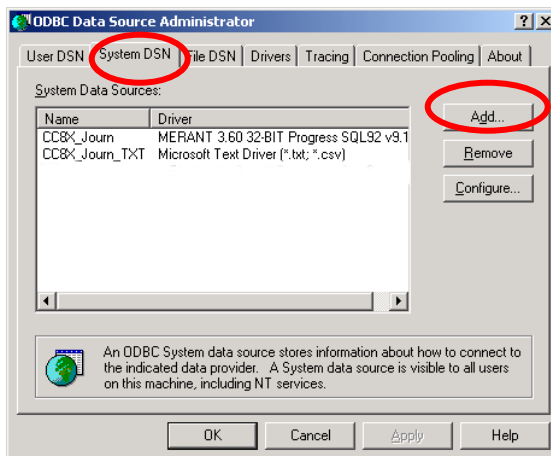


Figure 5: *ODBC Data Source Administrator* Window

4. Select the *System DSN* tab.
5. Click **Add**. The *Create New Data Source* window appears with a list of installed drivers.

6. Select the **MERANT 3.60 32-BIT Progress SQL92 v9.1D** driver from the Create New Data Source window and click **Finish**.

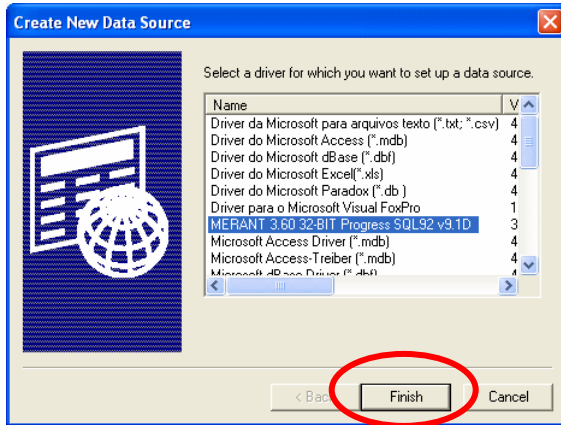


Figure 6: Select MERANT 3.60 32-BIT Progress SQL92 Driver

7. The *ODBC Progress SQL92 Driver Setup* window opens. Refer to the next section to set up the **SQL92** driver.

Setting up the SQL92 Driver

1. The *ODBC Progress SQL92 Driver Setup* window opens when you click *Finish* from the Create New Data Source window.
2. On the **General** tab, fill in the SQL92 Driver fields, as shown below and in Figure 7.

<i>Data Source Name:</i>	CFSRV (User-defined name of data source configuration , for example, CFSRV or any other name.)
<i>Description:</i>	Optional description
<i>Host Name:</i>	Server where the database is located. You can enter localhost if the database is on the current server
<i>Port Number:</i>	CFSRV (System port number, or service name for database listener process.)
<i>Database Name:</i>	CF (Name of database to which you want to connect)
<i>User ID:</i>	Login ID (User Name) to connect to the Database, such as <i>sysprogress</i> or <i>odbcuser</i>

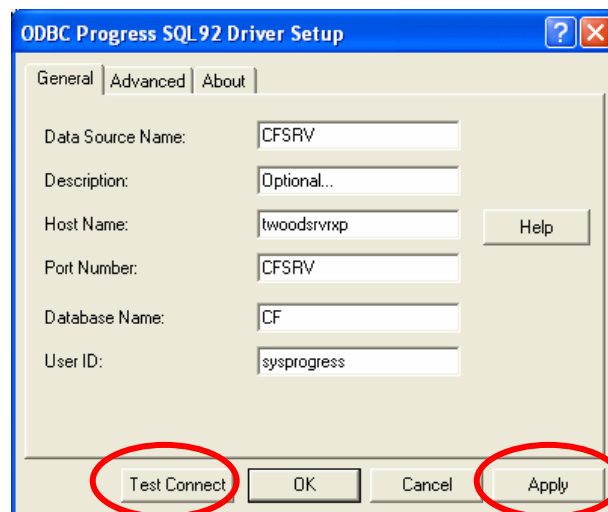


Figure 7: Sample ODBC Progress SQL 92 Driver Setup Window

3. Click **Test Connect** to try to connect to the data source using the connection properties you configured.
4. If the ODBC driver is able to connect to the data source, it displays a *Connection Established!* message.
5. Click **OK**. The *ODBC Progress Driver Setup* window reappears.
6. Click **Apply**.

Advanced Driver Settings

1. Click the **Advanced** tab to access the *Advanced SQL92* settings.
2. On the *Advanced* tab, select **READ UNCOMMITTED** in the *Default Isolation Level* field to prevent record locking.

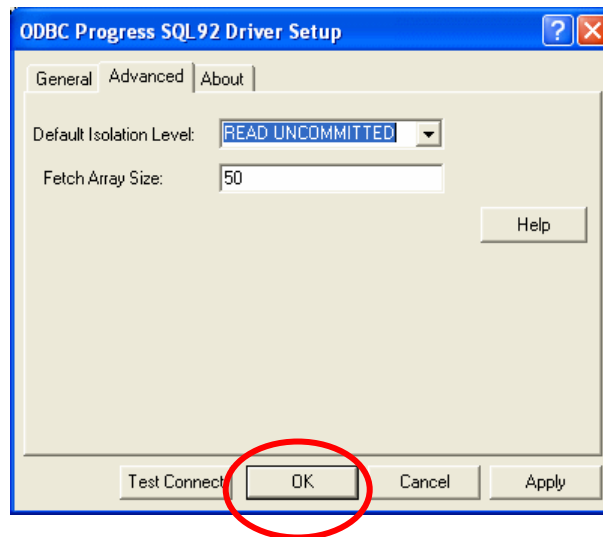


Figure 8: Set READ UNCOMMITTED on the SQL92 Driver Advanced Tab

3. Click **OK** to close the *ODBC Progress Driver Setup* window and set the values you entered as the default values when you connect to the data source.
4. The active data source appears in the ODBC Data Source Administrator window accessed from the *Windows Control Panel > Administrative Tools*.

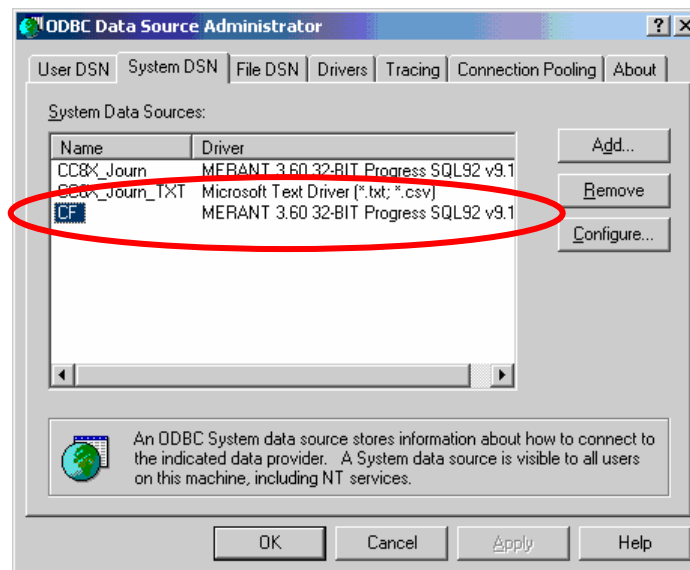


Figure 9: Active Data Source in ODBC Data Source Administrator Window

5. If necessary, you can change the default values using this same procedure to reconfigure your data source.

You can also override the default values by connecting to the data source using a connection string with alternative values.

Important! To prevent record locking, set **READ UNCOMMITTED** in the *Default Isolation Level* field on the SQL92 Driver Advanced Tab. Do **not** override the READ UNCOMMITTED default setting.

Using Progress SQL Explorer

SQL Explorer is a tool you can use to execute SQL-92 statements interactively. This section describes how to accomplish the following tasks with SQL Explorer:

- Start / Exit SQL Explorer
- Connect to a database with SQL Explorer
- Disconnect from a database with SQL Explorer
- Use SQL Explorer to:
 - Execute SQL statements
 - Grant permission on tables to regular users
 - Check database security and table access

For more information on using the SQL Explorer tool, refer to the SQL Explorer Online help.

Granting ODBC Access Rights to Users

In the Progress SQL-92 database, users do not have access to database objects until they are granted proper permissions.

Important The database must be running independent of C•CURE 800/8000. Shut down C•CURE 800/8000, then restart the database in single user mode from the Windows command shell.

The C•CURE 800/8000 Driver, 4GL applications such as the Administration application, or other SQL clients cannot be connected to the database when you GRANT or REVOKE SQL permissions. GRANT and REVOKE operations lock the database schema and require exclusive access.

Shut down C•CURE 800/8000 before Granting ODBC Privileges

1. Before shutting down the system, be sure to create ODBC user accounts as described in preceding sections.
 2. Terminate all connections to the database before issuing GRANT commands. As the C•CURE 800/8000 driver maintains a continuous connection to the databases, the driver must be shut down, and the database servers started up independently.
 3. Run the C•CURE 800/8000 **Full Shutdown** procedure.
-

Warning Shutting down the database will shut down the C•CURE 800/8000 Access Control system!

Starting the Database Server(s)

Start the database server(s) independent of C•CURE 800/8000. SYSPROGRESS should be the only user and the server(s) should not be running.

1. Select *Start>Run>cmd* to open a Windows command shell.
2. Enter these commands to start the CF (Configuration) database.

Commands to start the CF Database	Description
cd/d %cc800root%\4GL	(Change to database directory)
del database*.lk	(Delete any locked files.)
%DLC%\bin_mprosrv.exe -pf cfsrv.pf -db database\CF -N TCP -S CFSRV	(Start the CF database outside of CCure 800/8000.)

```

C:\>
C:\>
C:\>cd /d %cc800root%\4gl
D:\CCURE800\4gl>del database\*.lk
Could Not Find D:\CCURE800\4gl\database\*.lk
D:\CCURE800\4gl>%DLC%\bin\_mprosrv.exe -pf cfsrv.pf -db database\CF -N TCP -S CFSRV
14:54:46 BROKER : This broker will terminate when session ends. <5405>
14:54:46 BROKER 0: Multi-user session begin. <333>
14:54:46 BROKER 0: Begin Physical Redo Phase at 0 . <5326>
14:54:46 BROKER 0: Physical Redo Phase Completed at blk 85 off 3177 upd 0. <7161>
14:54:46 BROKER 0: Started for CFSRV using TCP, pid 3912. <5644>
D:\CCURE800\4gl>_
    
```

Figure 10: Commands to Start the CF (Configuration) Database

3. Enter these commands to start the Journal database.

Commands to start the Journal Database	Description
cd/d %cc800root%\4GL	(Change to database directory)
del database*.lk	(Delete any locked files)
%DLC%\bin_mprosrv.exe -db database\JL_00001 -N TCP -S JNSRV1	(Start the Journal database outside of CCure 800/8000, ex., JL_00001)

```

C:\>
C:\>cd /d %cc800root%\4gl
D:\CCURE800\4gl>del database\*.lk
Could Not Find D:\CCURE800\4gl\database\*.lk
D:\CCURE800\4gl>%DLC%\bin\_mprosrv.exe -db database\JL_00001 -N TCP -S JNSRV1
15:02:16 BROKER : This broker will terminate when session ends. <5405>
15:02:16 BROKER 0: Multi-user session begin. <333>
15:02:16 BROKER 0: Begin Physical Redo Phase at 64 . <5326>
15:02:16 BROKER 0: Physical Redo Phase Completed at blk 212 off 1077 upd 8232. <7161>
15:02:16 BROKER 0: Started for JNSRV1 using TCP, pid 1416. <5644>
D:\CCURE800\4gl>_
    
```

Figure 11: Commands to Start a Journal Database

Using SQL Explorer to Execute SQL92 Statements

Open SQL Explorer to execute SQL92 statements. Be sure the database is running in single user mode before executing the SQL statements.

Use the SYSPROGRESS login or another user account with DBA access to run the SQL Explorer application and connect to the database.

Starting the SQL Explorer GUI

1. From Windows Explorer, double-click on the file `\ccure800\4GL\SQL-EXP.BAT`.
2. The *SQL Explorer* window opens.

Connecting to the CF database from SQL Explorer

1. On the *File* menu, choose **Connect**. The *Connect* window opens.
2. Fill in the **Connect** window fields:

<i>Host Name:</i>	localhost (if the database is on the current server) or the name of the database server.
<i>Service or Port Number:</i>	CFSRV or System port number.
<i>Database Name:</i>	CF (Name of database to which you want to connect). Do not include path.
<i>User ID:</i>	SYSPROGRESS
<i>Password:</i>	Defined in CCURE800 for SYSPROGRESS

3. Click **OK**

Executing SQL Statements

1. Open the *SQL Explorer* window.
 2. Type the SQL statement in the top pane.
 3. Click the **Execute** button.
 4. SQL results appear in the bottom pane.
-

Caution In SQL Explorer, the **AutoCommit** option is false by default. Therefore, when the user disconnects, the changes made to the database are rolled back unless you override the default setting.

To make changes permanent, include a **COMMIT;** statement.

Important! Do **not** grant *ALL* or *UPDATE* access to any tables in the C•CURE 800/8000 database. These privileges permit write access to the database and could potentially enable users to corrupt data and make the system unusable.

Grant *SELECT* (Read Only) privileges to ODBC users.

SQL Statements to Grant and Revoke SELECT Access

Execute the following SQL statements to grant and revoke SELECT access to the pub.person table for each user with ODBC read access.

1. To grant SELECT access to the ODBC, use the GRANT statement:

```
GRANT SELECT ON PUB.person TO odbcuser;
COMMIT;
```

2. To revoke privileges given by the grant statement, use the REVOKE statement:

```
REVOKE SELECT ON PUB.person FROM odbcuser;
COMMIT;
```

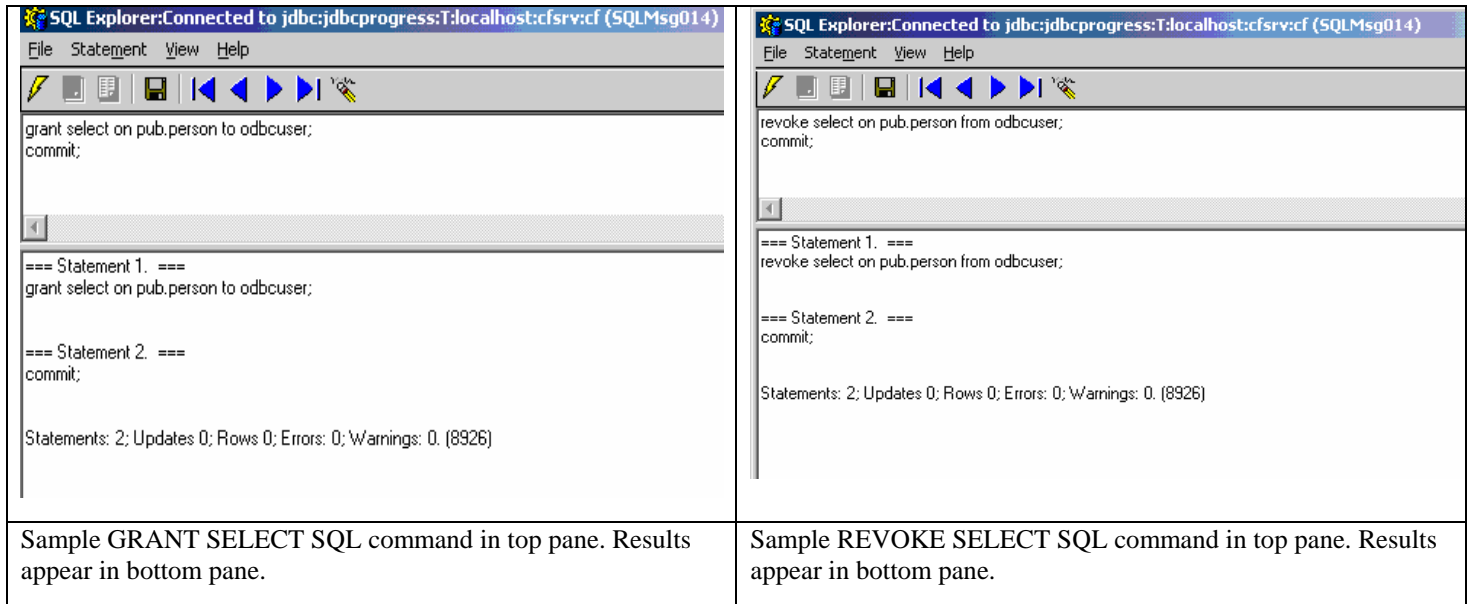


Figure 12: SQL Statements to GRANT and REVOKE SELECT Privileges

SQL Statements to Query the Configuration Database

This section contains sample SQL Statements to query data in the Configuration database:

Query Number of Regular Clearances

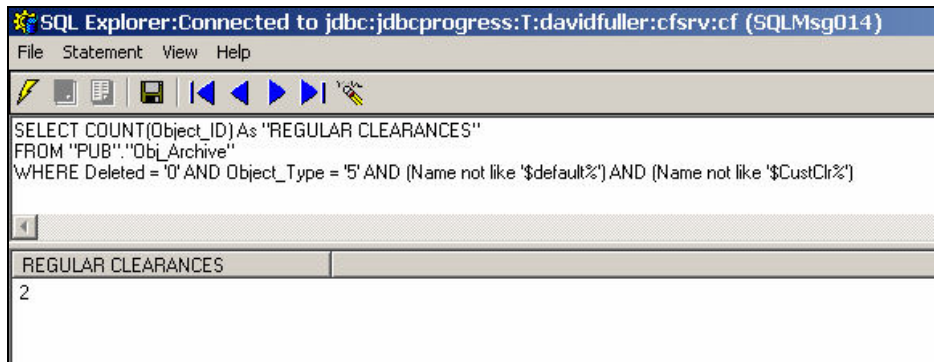


Figure 13: Number of Regular Clearances

Query Number of Custom Clearances

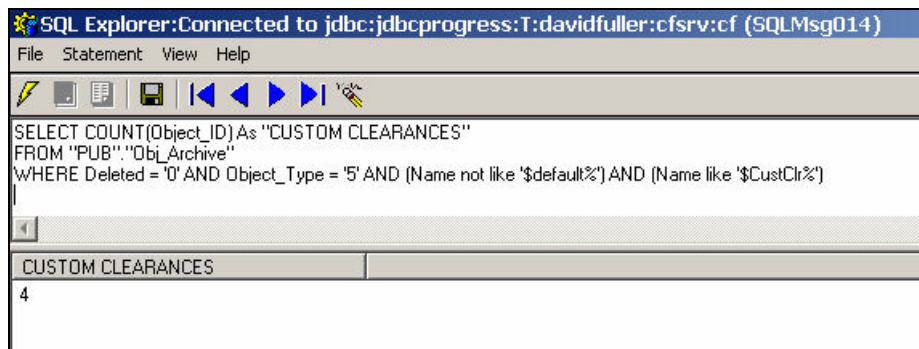


Figure 14: Number of Custom Clearances

Query Number of Existing Personnel Records

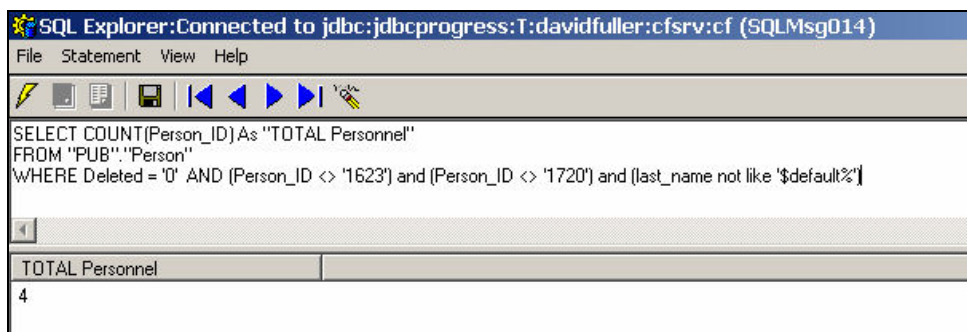


Figure 15: Number of Personnel Records

Query Number of Configured Events

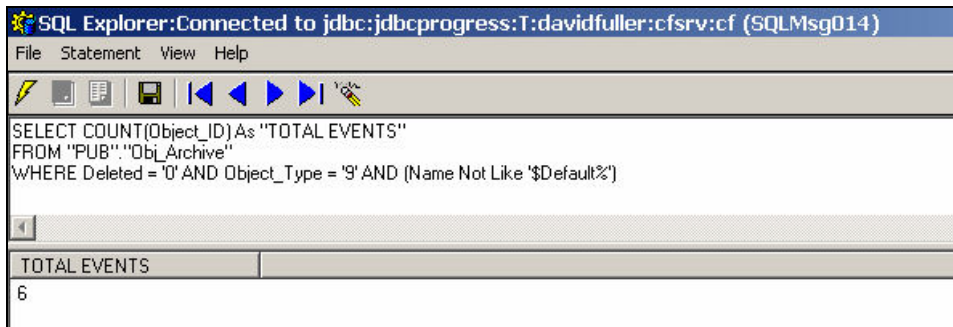


Figure 16: Number of Events

Query C•CURE800 System Serial Number

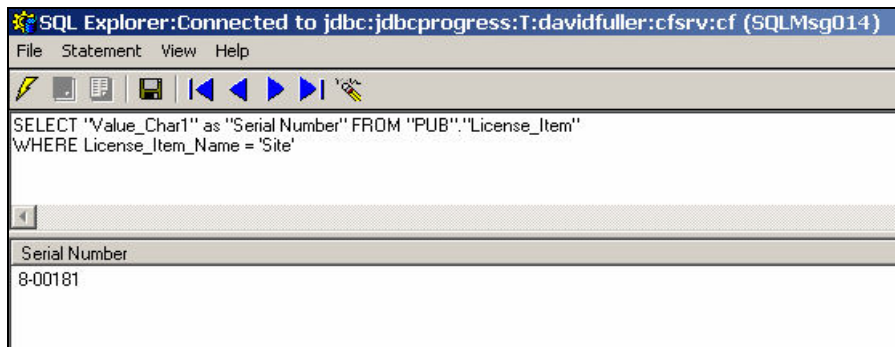


Figure 17: System Serial Number

Query Number of Licensed Badging Stations

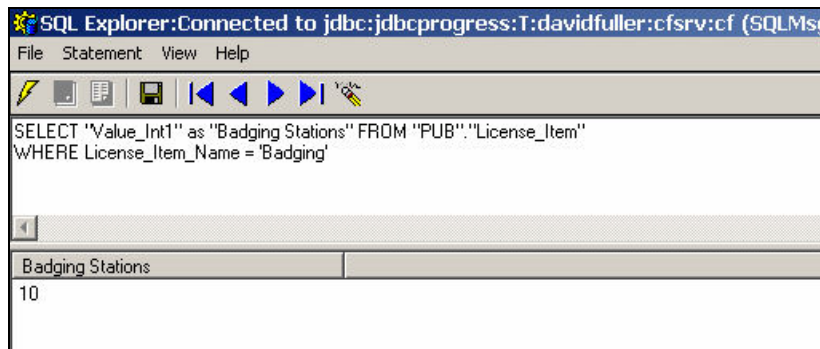


Figure 18: Number of Licensed Badging Stations

Query Objects in Configuration Database

SQL Explorer: Connected to jdbc:jdbcprogress:T:davidfuller:cfsrv:cf (SQLMsg014)

File Statement View Help

SELECT * from "pub"."obj_archive"

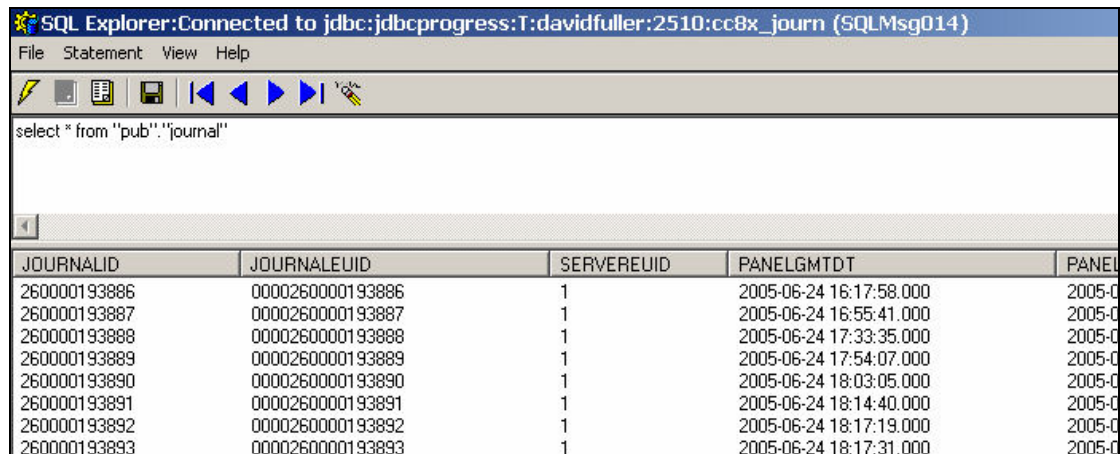
Object_ID	Name	Deleted	Object_Type	Partition_ID
1606	\$Default comm port	0	6	1701
1607	\$Default door	0	7	1701
1609	\$Default event	0	9	1701
1611	\$Default holiday	0	11	1701
1613	\$Default input	0	13	1701
1615	\$Default map	0	15	1701
1616	\$Default output	0	16	1701
1618	\$Default reader	0	18	1701
1620	\$Default time spec	0	20	1701
1621	\$Default area	0	21	1701
1624	\$Default NULL ApC	1	7	1701
1625	\$Default door report	0	22	1701
1626	\$Default apC report	0	22	1701
1627	\$Default input report	0	22	1701
1628	\$Default output report	0	22	1701
1629	\$Default person report	0	22	1701
1630	\$Default reader report	0	22	1701
1631	\$Default event report	0	22	1701
1632	\$Default layout	0	25	1701
1633	\$Default node	0	26	1701
1634	\$Default elevator	0	27	1701
1636	\$Default floor	0	29	1701
1638	Holiday List 1	0	12	1701
1639	Holiday List 2	0	12	1701
1640	\$Unlisted	0	26	1701
1642	\$Default elevator report	0	22	1701
1643	\$Default floor report	0	22	1701
1644	\$Default area report	0	22	1701
1645	\$Default CCTV report	0	22	1701
1646	\$Default CCTV com report	0	22	1701
1647	\$Default CCTV pro report	0	22	1701
1648	\$Default modem report	0	22	1701
1651	\$Default CCTV Switcher	0	32	1701
1652	\$Default CCTV action	0	33	1701
1653	\$Default CCTV proto	0	34	1701
1654	\$Default host modem	0	35	1701
1657	\$Default Admin priv	0	36	1701
1658	\$Default Moni priv	0	37	1701
1659	\$Full Administration	0	36	1701

Figure 19: Configuration Database Contents

SQL Statements to Query the Journal Database

This section contains sample SQL Statements to query data in the Journal database:

Query Number of Records in Journal Database



SQL Explorer: Connected to jdbc:jdbcprogress:T:davidfuller:2510:cc8x_journ (SQLMsg014)

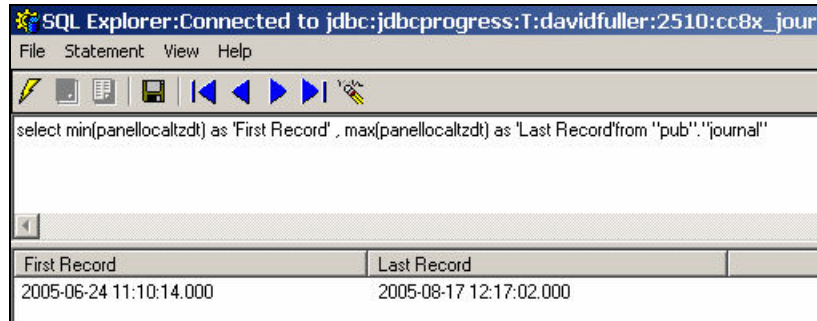
File Statement View Help

select * from "pub"."journal"

JOURNALID	JOURNALEUID	SERVEREUID	PANELGMTDT	PANEL
260000193886	0000260000193886	1	2005-06-24 16:17:58.000	2005-0
260000193887	0000260000193887	1	2005-06-24 16:55:41.000	2005-0
260000193888	0000260000193888	1	2005-06-24 17:33:35.000	2005-0
260000193889	0000260000193889	1	2005-06-24 17:54:07.000	2005-0
260000193890	0000260000193890	1	2005-06-24 18:03:05.000	2005-0
260000193891	0000260000193891	1	2005-06-24 18:14:40.000	2005-0
260000193892	0000260000193892	1	2005-06-24 18:17:19.000	2005-0
260000193893	0000260000193893	1	2005-06-24 18:17:31.000	2005-0

Figure 20: Number of Records in Journal Database

Query First and Last Journal Records based on Date/Time



SQL Explorer: Connected to jdbc:jdbcprogress:T:davidfuller:2510:cc8x_journ

File Statement View Help

select min(panellocaltzt) as 'First Record' , max(panellocaltzt) as 'Last Record' from "pub"."journal"

First Record	Last Record
2005-06-24 11:10:14.000	2005-08-17 12:17:02.000

Figure 21: Date Range of Records in Journal Database

Query All Card ADMIT Records in Journal Database

SQL Explorer: Connected to jdbc:jdbcprogress:T:davidfuller:2510:cc8x_journ (SQLMsg014)

File Statement View Help

```
select 'panellocaltzt', 'messagetext' from 'pub'. 'journal'
WHERE messagecodeIDX = 1002
ORDER BY panellocaltzt
```

PANELLOCALTZDT	MESSAGETEXT
2005-06-24 13:54:07.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-24 14:03:05.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In] [Unused]
2005-06-24 14:14:40.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at APC-1 MAG STRIPE DOOR [In]
2005-06-24 14:17:31.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at APC-1 MRM DOOR [In]
2005-06-24 14:18:11.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at Lite Door 1 [In]
2005-06-24 15:14:54.000	Admitted Fuller, David (Card #14242) at Lite Door 1 [In]
2005-06-27 07:16:17.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-27 07:17:45.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-27 08:19:04.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-27 08:26:17.000	Admitted Fuller, David (Card #14242) at iSTAR II - Door 1 [In]
2005-06-27 08:34:58.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at APC-1 MRM DOOR [In]
2005-06-27 08:38:20.000	Admitted Fuller, David (Card #14242) at iSTAR II - Door 1 [In]
2005-06-27 08:38:51.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at iSTAR II - Door 2 [In]
2005-06-27 09:14:02.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at APC-1 MAG STRIPE DOOR [In]
2005-06-27 10:18:28.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-27 10:20:54.000	Admitted \$\$Del_Prs(90532,11) (Card #11) at APC-1 MAG STRIPE DOOR [In]
2005-06-27 10:32:55.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-27 15:07:36.000	Admitted Armpit, Hairy Nasty (Card #616) at iSTAR II - Door 2 [In]
2005-06-27 15:09:03.000	Admitted Armpit, Hairy Nasty (Card #616) at APC-1 MAG STRIPE DOOR [In]
2005-06-27 15:10:15.000	Admitted Fuller, David (Card #14242) at iSTAR II - Door 1 [In]
2005-06-27 15:17:14.000	Admitted Armpit, Hairy Nasty (Card #616) at APC-1 MRM DOOR [In]
2005-06-27 15:17:22.000	Admitted Schiavoneowski, Luigi (Card #666) at APC-1 MAG STRIPE DOOR [In]
2005-06-27 15:23:11.000	Admitted Schiavoneowski, Luigi (Card #666) at iSTAR II - Door 2 [In]
2005-06-28 07:06:08.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-28 07:30:51.000	Admitted Fuller, David (Card #14242) at iSTAR II - Door 1 [In]
2005-06-28 08:00:17.000	Admitted Fuller, David (Card #14242) at APC-1 MRM DOOR [In]
2005-06-28 09:02:19.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-28 09:42:09.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]
2005-06-28 14:57:10.000	Admitted Fuller, David (Card #14242) at Lite Door 1 [In]
2005-06-28 14:57:22.000	Admitted Fuller, David (Card #14242) at APC-2 HID DOOR [In]
2005-06-28 14:59:31.000	Admitted Fuller, David (Card #14242) at APC-2 HID DOOR [In]
2005-08-17 13:12:35.000	Admitted Fuller, David (Card #14242) at APC-1 PROX DOOR [In]

Figure 22: Card Admit Records in Journal Database

Disconnect from the Database and Exit SQL Explorer

After executing SQL commands, disconnect from the database and exit the SQL Explorer.

1. On the File menu, select **Disconnect** to disconnect from the database.
2. On the File menu, select **Exit** to exit SQL Explorer.

Shut down and Restart the C•CURE 800/8000 Database Server

After disconnecting from the database and exiting SQL Explorer, shut down and restart the database server(s) for the C•CURE 800/8000 databases.

1. Open a Windows command shell.
2. Enter these commands to shut down the CF database:

Commands to Shut Down the CF Database	Description
<code>cd/d %cc800root%\4gl</code>	(Change to the database directory)
<code>%DLC%\bin_mprshut -db database\cf</code>	(Shut down the CF database)

```

C:\>
C:\>cd /d %cc800root%\4gl
D:\CCURE800\4gl>%DLC%\bin\_mprshut -db database\cf

          1  Disconnect a User
          2  Unconditional Shutdown
          3  Emergency Shutdown (Kill All)
          x  Exit

Enter choice> 2
Shutdown is executing. (1613)
Shutdown complete. (1614)
    
```

Figure 23: Commands to Shut down the CF Database

3. Enter these commands to shut down the Journal database:

Commands to Shut Down the Journal Database	Description
cd/d %cc800root%\4GL	(Change to the database directory)
%DLC%\bin_mprshut -db database\jl_00001	(Shut down the Journal_00001 database).

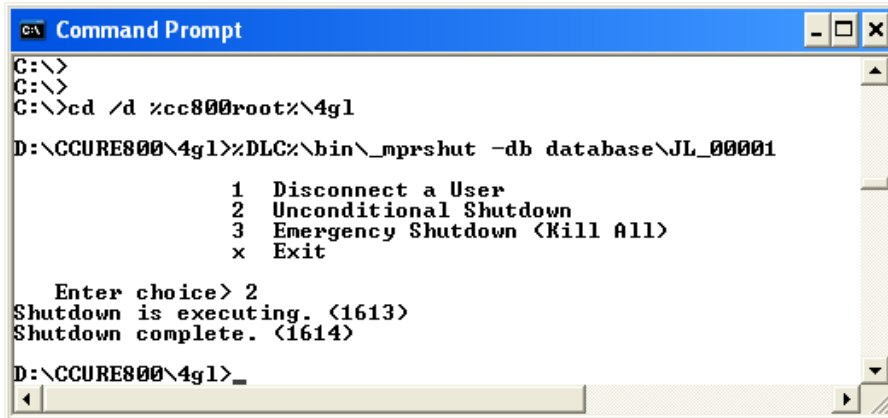


Figure 24: Commands to Shut down a Journal Database

4. Run the C•CURE 800/8000 **Full Startup** procedure to restart the C•CURE 800/8000.

SQL Problems and Solutions

This section describes some possible problems and solutions when issuing SQL statements.

The PUB schema prefix is not specified for a 4GL table.

Solution: Prefix the name of the table, view, or synonym with PUB.

Incorrect: SELECT * FROM person

Change to: SELECT * FROM PUB.person

The specified name has hyphens.

Solution: Enclose the hyphenated name in double quotes.

Incorrect: SELECT * FROM PUB.Ref-Call

Change to:: SELECT * FROM PUB."Ref-Call"

The SYSPROGRESS prefix is not specified for a SQL-92 system table or view.

Solution: Prefix the name of the table or view with SYSPROGRESS.

Incorrect: SELECT * FROM systables

Correct: SELECT * FROM SYSPROGRESS.systables

The specified table, view, or synonym is not available in the current schema

Solution: Prefix the object with the correct schema name or use the SET SCHEMA statement to change the current one.

Correct:

SET schema 'pub'

The specified table, view or synonym does not appear to be available in the database

Solution: Use the correct object name, found in the database. If you are unsure of the correct object name, query the systables view to list all tables.

Correct:

SELECT * FROM SYSPROGRESS.systables

Contact Information

For system problems that you cannot resolve, contact your authorized Software House dealer.

Setting up the ODBC Client for the Journal Database

Setting up and configuring the ODBC data source and ODBC driver for the Journal database (JL) is similar to the configuration (CF) database.

Real-time information about card accesses and other recorded system activity is stored in the JOURN table of the Journal database, in a file called JL_XXXX.db, (where XXXX is the volume number of the journal).

The C•CURE 800/8000 driver automatically switches to a new journal file when the current journal is filled. Journal files are numbered sequentially, beginning with JL_00001.

When the journal database server is started, it is connected to one of two TCP/IP ports:

- Odd-numbered journal files connect to port *JNSRV1*.
- Even numbered journals connect to port *JNSRV2* as shown below:

Because of journal database switching, if you want to access the active journal database, you must update the ODBC Data Source for the journal whenever the journal switch occurs.

NOTE When C•CURE 800/8000 activates a new journal, the ODBC Data Source must be updated so that the *Database Name* references the new journal filename, and the *Port Number* references the new service name.

For example, the following screen shows an example of setting up an even numbered journal database (JN_00002) on port JNSRV2.

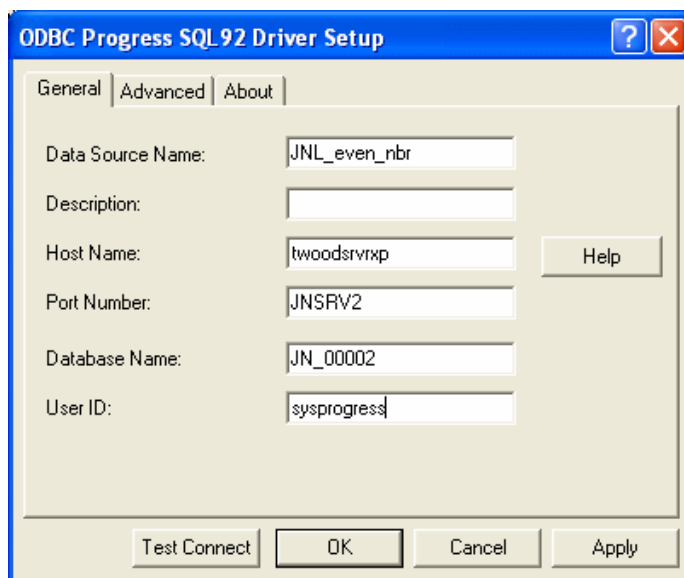


Figure 25: Setting up an Even-numbered Journal Database

Using ODBC with a Prior Journal

User names and ODBC passwords are created and maintained separately within each database. Each journal volume is a separate database.

- Any changes to user names and ODBC passwords are applied only to the CF and current Journal databases.
- Historical Journal databases have the user names and ODBC passwords that were in place at the time those journals were active.

Note You must recreate access rights in the new journal database for any ODBC users other than SYSPROGRESS.

If you change or remove a user name or ODBC password, you must record the old name or password and use it to access previous journal volumes.

-
- Once a journal switch occurs, only the SYSPROGRESS user continues to have access rights. While C•CURE 800/8000 propagates user names and passwords if you use the *Add/Edit Personnel* screen in the *Administration* option to create the users, C•CURE 800/8000 does not propagate the SQL92 ODBC user access rights into new journal databases.
 - You must recreate access rights in the new journal database for any ODBC users other than SYSPROGRESS.

You must also recreate access rights for the CF database whenever a database update occurs; for example, when you update to a new version of C•CURE 800/8000.

Starting and Stopping a Journal Server via a Batch File

ODBC cannot be used to connect to a database unless there is a Progress database server running for that database; only the current journal has a database server running. To use ODBC against a prior journal, you must start up a database server for that database

Starting a Journal Server via a Batch File

To start a Progress database server for a specific journal database, create and run a batch file similar to the one below called *STARTJNL.BAT*.

In the sample batch file below, a database server is started as a service called JNSRV3 for the journal database specified by the *%I%* variable (the first parameter after the command, for example, JL_00009 or JL_00001).

Make sure that you do not use the services JNSRV1 or JNSRV2 because these services are used by C•CURE 800/8000 for the active journal.

```
REM      Start a journal server.
REM      The following batch file may be used to start up a
REM      database server for an archived journal database.
REM      You cannot start a database server that is already running.
REM      Run the stopjnl.bat batch procedure to stop the server before
REM      running the startjnl.bat batch procedure.
REM      When invoking this procedure, specify the name of the
REM      journal (e.g. JL_00009, JL_00001) you want to start
REM
REM      STARTJNL JL_00009
REM
setlocal
cd/d %cc800root%\4GL
%DLC%\bin\_mprosrv.exe -db database\%1% -N TCP -S JNSRV3
endlocal
```

Run the Start Journal batch file from the C•CURE 800/8000 server machine and specify the journal database.

For example, this command starts the JL_00001 database (which has been shut down).

```
STARTJNL JL_00001
```

```
C:\>
C:\>dir st*
Volume in drive C has no label.
Volume Serial Number is 1038-E7AD

Directory of C:\

08/17/2005  11:13 AM                101 startjnl.bat
08/17/2005  11:22 AM                 81 stopjnl.bat
             2 File(s)                182 bytes
             0 Dir(s)  7,815,221,248 bytes free

C:\>startjnl jl_00001
C:\>setlocal
C:\>cd /d d:\CCURE800\4GL
D:\CCURE800\4gl>d:\CCURE800DLC\bin\mprosrv.exe -db database\jl_00001 -N TCP -S JNSRU3
15:18:33 BROKER   : This broker will terminate when session ends. <5405>
15:18:33 BROKER   0: Multi-user session begin. <333>
15:18:33 BROKER   0: Begin Physical Redo Phase at 128 . <5326>
15:18:33 BROKER   0: Physical Redo Phase Completed at blk 212 off 1077 upd 5552. <7161>
15:18:33 BROKER   0: Started for JNSRU3 using TCP, pid 3984. <5644>
D:\CCURE800\4gl>endlocal
C:\>
```

Figure 26: Running the Start Journal Batch File

Stopping a Journal Server via a Batch File

To stop a Progress database for a particular journal database, create and run a batch file similar to the one below called *STOPJNL.BAT*.

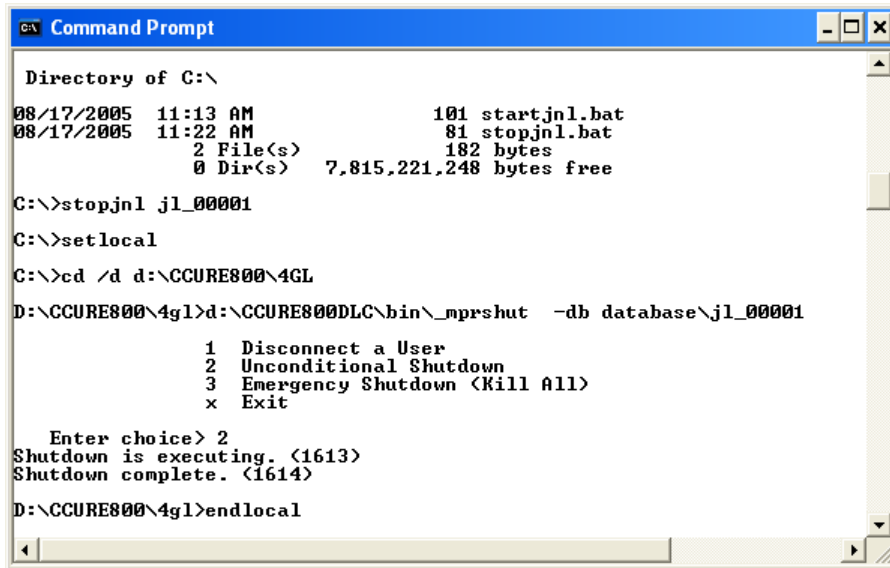
In the sample batch file below, the database server for the journal database specified by the first parameter (*%I%*) is shut down.

```
REM      Stop a journal server.
REM
REM      The following batch file may be used to stop up a database
REM      server for an archived journal database that was started using
REM      the startjnl.bat batch procedure.
REM      The name of the journal (e.g. JL_00003, or JL_00001) to stop
REM      must be supplied when invoking this procedure. e.g.:
REM
REM      STOPJNL JL_00009
REM
setlocal
cd/d %cc800root%\4GL
%DLC%\bin\_mprshut -db database\%1%
endlocal
```

Run the batch file from the C•CURE 800/8000 server machine and specify the journal database.

For example, this batch command stops the JL_00001 database:

STOPJNL JL_00001



```
Command Prompt
Directory of C:\
08/17/2005  11:13 AM                101 startjnl.bat
08/17/2005  11:22 AM                81 stopjnl.bat
             2 File(s)              182 bytes
             0 Dir(s)              7,815,221,248 bytes free

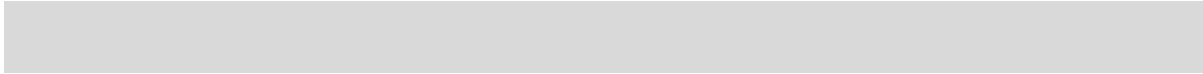
C:\>stopjnl jl_00001
C:\>setlocal
C:\>cd /d d:\CCURE800\4GL
D:\CCURE800\4gl>d:\CCURE800DLC\bin\_mprshut -db database\jl_00001

             1 Disconnect a User
             2 Unconditional Shutdown
             3 Emergency Shutdown <Kill All>
             x Exit

Enter choice> 2
Shutdown is executing. <1613>
Shutdown complete. <1614>
D:\CCURE800\4gl>endlocal
```

Figure 27: Running the Stop Journal Batch File

Chapter 3



Database Schema

This chapter describes the sub-schema (views) of the CF (Configuration) and JOURN (Journal) databases.

In This Chapter

- Overview of the Database Schema
- How the CF (Configuration) Database is Organized
- Table Layouts
- Database Schema Conventions
- Asset Object
- Clearance Object
- Door Object
- Groups Object
- Inputs Object
- Outputs Object
- Person Object
- Reader Object
- Time Spec Object
- Time Zone Object
- Unit Object
- Journal Object
- Reference Tables

How the CF (Configuration) Database is Organized

The C•CURE System operates on security objects such as readers, doors, and time specs. The CF database is organized in terms of these security objects, with several tables comprising a single object. In some cases, one table may be part of more than one object.

Each object contains a unique ID, which in most cases is used to join it to related tables. For example, the door object is uniquely identified by Door_ID. In a few cases, one table is joined to another using a cross-reference (link) table or programmatically using exception code.

Table Layouts

This chapter includes table layouts that can be accessed through the ODBC. Conventions used in deriving the schema are provided where appropriate.

Table relationship diagrams use the following notation:

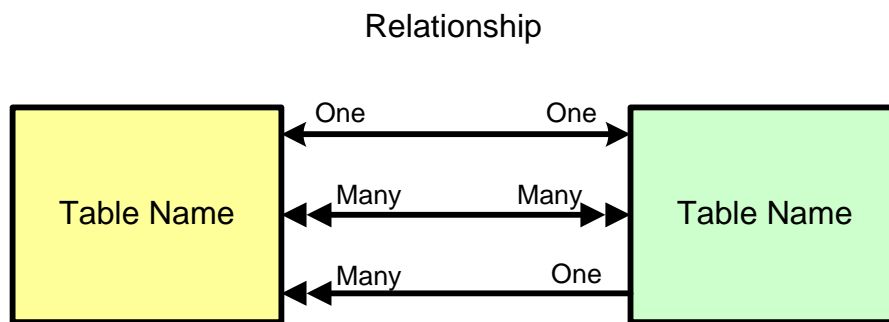


Figure 28: Database Schema Conventions

NOTE:

Descriptions of relations use the word “may” to indicate that a null set is allowed. For example, ‘a worker *may* be assigned to one or many projects’ allows the case where a worker is not assigned to any project.

The converse, “a worker *is* assigned to one or many projects” indicates all workers are assigned to at least one project.

Database Schema Conventions

The following conventions were used to generate the database schema:

- When possible, the primary unique index to a table was made to be a sequentially assigned integer. The name of this primary unique index field is the name of the table plus the characters “_id”.
- The name of the primary unique index to a table is assigned as follows:
 - for a composite index to a table which is a cross reference, it is the same name as the first field of the composite index
 - for a composite index to a table which is not a cross reference, the name of the table followed by the characters “_index”
 - for all other cases, the name of the index is the same name as the field being indexed.
- All primary unique index fields are defined as “Mandatory”. The application is prohibited from creating records where the primary unique key is unknown.
- Tables which contain data that can be joined use the same field names and index names for those fields which comprise the join.
- Field definitions are sized to reflect the value range for that field subject to the following general guidelines:
 - Numeric value definitions allow positive and negative values.
 - Integers are defined to accommodate the maximum range of the integer field (-2,147,483,648 to +2,147,483,647).
 - Decimal fields are defined with 12 positions *before* and 4 positions *after* the decimal point.
 - Character fields are no more than 3000 characters for fields normally viewed through an editor widget (to comply with the maximum field size supported by the Progress Data Administration data unload and reload utilities) and no more than 320 characters for non-editor fields (to comply with the Progress limitation of the maximum field size that can be displayed in a frame).
 - Character fields that are used as indexes are no more than 255 characters in length to be compliant with MS Access constraints when the database is accessed through ODBC
 - Date fields are defined in the database with a 4 digit year (ie: xx/xx/yyyy) and the separation character is a forward-slash. The representation of such fields can vary on a screen-by-screen basis.
 - Whenever a database table contains text data that may be stored in more than one language, the table will have a field (usually called Language_IDX) to indicate the language of the record string.

Following are the possible Language IDX values:

```

&GLOBAL-DEFINE PRM$LNG_RUSSIAN                25 /* Russian */
&GLOBAL-DEFINE PRM$LNG_CHINESE_TRADITIONAL 1028 /* Traditional Chinese */
&GLOBAL-DEFINE PRM$LNG_GERMAN                1031 /* German */
&GLOBAL-DEFINE PRM$LNG_ENGLISH              1033 /* English */
&GLOBAL-DEFINE PRM$LNG_SPANISH              1034 /* Spanish */
&GLOBAL-DEFINE PRM$LNG_FRENCH               1036 /* French */
&GLOBAL-DEFINE PRM$LNG_CHINESE_SIMPLIFIED  2052 /* Simplified Chinese */
    
```

Asset Object

Definition

An asset is any physical item that the user wishes to be tracked by the system (such as a PC).

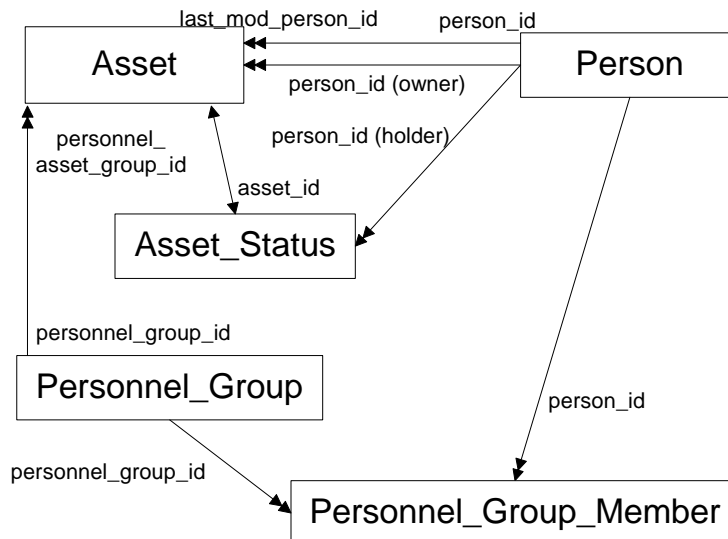
Relational Description

An Asset:

- May be directly related to one asset_status.
- May be directly related to one personnel_group.

A person:

- May be directly related to one or more assets (as the owner of that asset and/or as the person to last modify the asset record).
- May be directly related to one or more asset_status (as the person having physical possession of that asset).



Asset

Index	Fields	Type
Asset_ID	Asset_ID	primary
Asset_Name	Asset_Name	
Asset_Mfg_SN	Manufacturers_SN	unique
Asset_SN	Tag_SN	
Asset_Tag	Facility_ID	
	Tag_Number	
	Issue_Code	

Order	Field Name	Data Type	Flags	Format
10	Asset_ID	inte	im	->>>>>>>>9
20	Asset_Name	char	i	X(20)
30	Facility_ID	inte	i	>>>>>>>9
40	Tag_Number	inte	i	>>>>>>>9
50	Issue_code	inte	i	>>>>>>>9
60	Description	char		X(3000)
70	Tag_SN	char	i	X(20)
80	Manufacturers_SN	char	i	X(20)
90	Asset_Category_ID	inte		->>>>>>>9
100	Time_Spec_ID	inte		->>>>>>>9
110	Personnel_Group_ID	inte		->>>>>>>9
120	Asset_type	inte		>9
130	Person_ID	inte		->>>>>>>9
140	Owner_Allowed	logi		yes/no
150	Group_Allowed	logi		yes/no
160	Everyone_Allowed	logi		yes/no
170	Aquisition_Date	date		99/99/9999
180	Warranty_Date	date		99/99/9999
190	Maintenance_date	date		99/99/9999
220	Duration	inte		>>9
230	Assigned_Area_ID	inte		->>>>>>>9
240	Last_Mod_Datetime	inte		->>>>>>>9
250	Last_Mod_Person_ID	inte		->>>>>>>9
260	Text1	char		X(255)
270	Text2	char		X(255)
280	Text3	char		X(255)
290	Deleted	logi		yes/no
300	Noticed	logi		yes/no
310	Last_Mod_Object_ID	inte		->>>>>>>9
311	Lost	logi		yes/no
320	Report_Overdue	logi		yes/no

Asset_Status

Index	Fields	Type
Asset_ID Due_DT	Asset_ID Overdue_Logged Due_DT	primary

Order	Field Name	Data Type	Flags	Format
10	Asset_ID	inte	i	->>>>>>>>9
20	Last_Reader_ID	inte		->>>>>>>>9
30	Last_Area_ID	inte		->>>>>>>>9
40	Person_ID	inte	m	->>>>>>>>9
50	Due_DT	inte	i	>>>>>>>>9
60	Checkout_DT	inte		>>>>>>>>9
70	Overdue_Logged	logi	i	yes/no

Internal References

Table	Method	Phrase(s)
Users	Join - Person_ID	<ul style="list-style-type: none"> • users of person • person of users
Clear_Person	Join - Person_ID	<ul style="list-style-type: none"> • clear_person of person

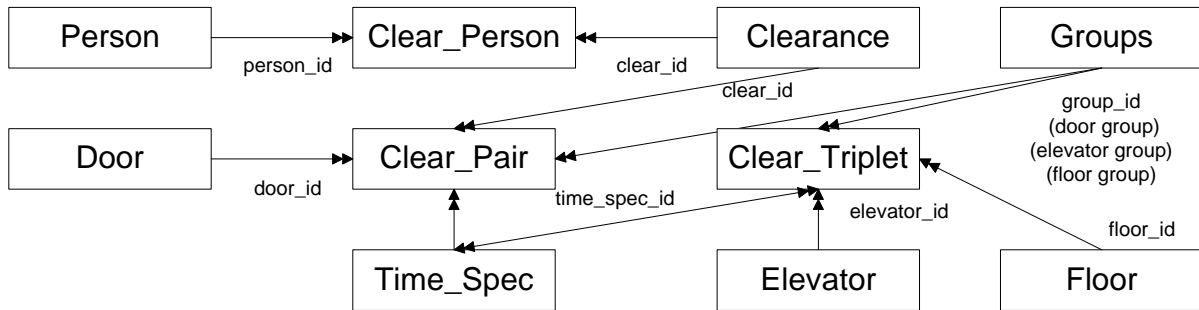
Clearance Object

Business Definition

An indication of whether or not a person is allowed to pass through a door.

Relational Description

- A clearance consists of one or many clearance pairs or clearance triplets. Each clearance pair references a time spec and either a door or a door group. Each clearance triplet references a time spec, an elevator or elevator group, and a floor or floor group.
- Clearances may be related to one or several persons through the clear_person table.



Primary Tables

Clearance

Index	Fields	Type
Clear_ID	Clear_ID	primary unique
Clearance_Name	Clearance_Name	unique
Clearance_EUID	Clearance_EUID	unique

Order	Field Name	Data Type	Flags	Format
10	Clear_ID	inte	im	->>>>>>>9
20	Clearance_Name	char	i	X(50)
30	Description	char		X(3000)
50	Partition_ID	inte	im	->>>>>>>9
60	Clearance_EUID	char	i	X(40)
70	Activation_DateTime	inte		->>>>>>>9
80	Expiration_DateTime	inte		->>>>>>>9
90	Use_Activation_Date	logi		yes/no
100	Use_Expiration_Date	logi		yes/no
110	Is_Custom	logi		yes/no
120	Person_ID	inte	i	->>>>>>>9
130	Is_Global	logi		yes/no
140	Is_Template	logi		yes/no

Foreign References

Table	Method	Phrase(s)
Person	Clear_Person Table	<ul style="list-style-type: none"> • each clearance of clear_person, each clear_person of person • each person of clear_person, each clear_person of clearance
Time_Spec	Clear_Pair Table	<ul style="list-style-type: none"> • each time_spec of clear_pair, each clear_pair of clearance • each clearance of clear_pair, each clear_pair of time_spec
	Clear_Triplet Table	<ul style="list-style-type: none"> • each time_spec of clear_triplet, each clear_triplet of clearance
Door	Clear_Pair Table	<ul style="list-style-type: none"> • each clear_pair of clearance, each door where door.door_id = clear_pair.object_id • each clearance of clear_pair where clear_pair.object_id = door.door_id
Elevator	Clear_Triplet Table	<ul style="list-style-type: none"> • each clear_triplet of clearance, each elevator where elevator.elevator_id = clear_triplet.elevator_or_grp_id
Floor	Clear_Triplet Table	<ul style="list-style-type: none"> • each clear_triplet of clearance, each floor where floor.floor_id = clear_triplet.floor_or_grp_id
Groups (door)	Clear_Pair Table	<ul style="list-style-type: none"> • each clear_pair of clearance, each groups where groups.group_id = clear_pair.object_id • each clearance of clear_pair where clear_pair.object_id = groups.group_id • each groups where groups.group_type = {&prm\$obj_door_group}
Groups(elevator)	Clear_Triplet Table	<ul style="list-style-type: none"> • each clearance of clear_triplet where clear_triplet.elevator_or_grp_id = group.group_id • each group where groups.group_type = {prm\$obj_elevator_group} • each clearance of clear_triplet where clear_triplet.elevator_or_grp_id = group.group_id
Groups(floor)	Clear_Triplet Table	<ul style="list-style-type: none"> • each groups where groups.group_type = {prm\$obj_floor_group}
Users	Clear_Person Table	<ul style="list-style-type: none"> • each users of clear_person

Internal References

Method	Phrase(s)
Join - Clear_ID	<ul style="list-style-type: none">• clear_pair of clearance• clear_person of clearance• clear_triplet of clearance
Join - Time_Spec_ID	<ul style="list-style-type: none">• clear_pair of time_spec• clear_triplet of time_spec
Join - Person_ID	<ul style="list-style-type: none">• clear_person of person• clear_person of users

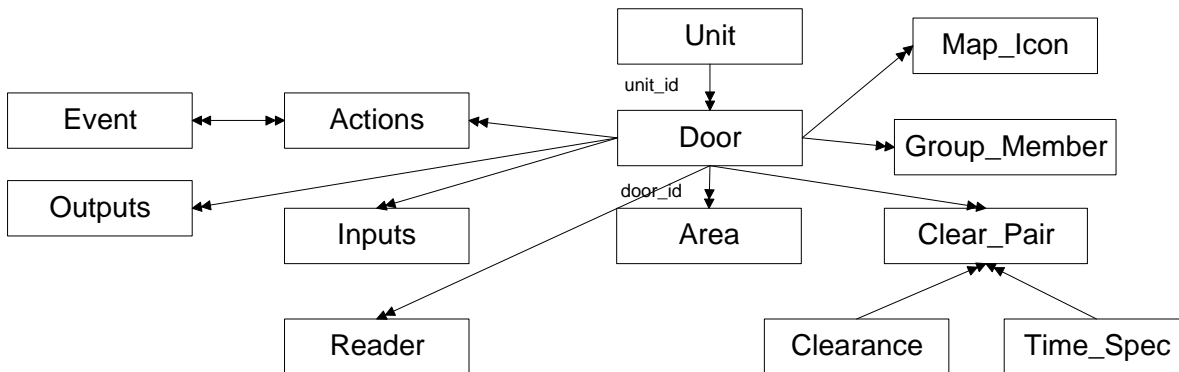
Door Object

Business Definition

A point of passage between two areas through which access may be monitored or controlled.

Relational Description

- A door is a stand-alone object that will appear on the guard station only when it has a relationship to a reader or a door latch relay (output).
- May be directly related to one or two inputs which may be DSM (&prm\$ost_door_switch_monitor) or RTE (&prm\$ost_request_to_exit).
- May be directly related to one or two outputs which may be DLR (&prm\$ost_unlocked) or ADA (&prm\$ost_ada_unlocked).
- May be directly related to one or two readers which may be &prm\$ost_reader_1 or &prm\$ost_reader_2.
- May be related to one or many events through the Actions table where the door is the source_id and the state_idx field varies.
- May belong to a group.
- May be related to one or many clearances through the clear_pair table.
- May be related to one or many maps through the map_icon table.
- May be the target of an event through the actions table.
- May be related to 0 or two areas based on the entrance_area_id and exit_area_id
- May be controlled by InfoProx reader. The major differences from the standard C•Cure door are the second swipe action, the extra timers and the extra time specifications.



Primary Tables

Door

Index	Fields	Type
Door_ID	door.door_id	primary unique
Door_Name	door_name	unique
Unit_ID	door.unit_id	active
Entrance_Area_ID	entrance_area_id	active
Exit_Area_ID	exit_area_id	active
RF_controller_ID	RF_controller_ID	active
Control_Zone_ID	Control_Zone_ID	active
Mapping_Number	Mapping_Number	unique
Partition_ID	Partition_ID	active

Order	Field Name	Data Type	Flags	Format
10	Door_ID	inte	im	->>>>>>>>>9
20	Unit_ID	inte	i	->>>>>>>>>9
30	Door_Has_RTE	logi		yes/no
40	Unlock_Door_on_RTE	logi		yes/no
50	Partition_ID	inte	im	->>>>>>>>>9
51	Shunt_DSM_on_RTE	logi		yes/no
60	Continuously_Active	logi		yes/no
70	Relock_After_Open	logi		yes/no
80	Delay_Relock_Time	inte		>>9
90	Unlock_Time	inte		>>9
100	Shunt_Time	inte		>>9
110	Door_Name	char	i	X(50)
120	Description	char		X(3000)
130	Entrance_Area_ID	inte	i	->>>>>>>>>9
140	Exit_Area_ID	inte	i	->>>>>>>>>9
150	Asset_Entrance_Area_ID	inte		->>>>>>>>>9
160	Asset_Exit_Area_ID	inte		->>>>>>>>>9
180	Asset_Card_Link_Time	inte		>, >>9
200	In_Directional_Link_Time	inte		>, >>9
220	Out_Directional_Link_Time	inte		>, >>9
230	RF_Controller_ID	inte	im	->>>>>>>>>9
240	Mapping_Number	inte	i	>>>9
250	Asset_Unit_ID	inte		->>>>>>>>>9
260	Control_Zone_ID	inte	i	->, >>>, >>9
270	Annunciate	logi		yes/no
280	Unlock_on_CZ_Access	logi		yes/no
290	Alternate_Shunt_Time	inte		>>>>9
300	Shunt_Exp_Relay_Time	inte		>>9
310	Max_Time_Display	inte		>>9
320	Asset_Link_Mode	inte		>9
330	Asset_Link_Order	inte		>9
340	ipx_Primary_Reader_is_Inbound	logi		yes/no
350	ipx_Uses_Secondary_Reader	logi		yes/no
360	Is_Infoprox	logi		yes/no

370	CZ_Card_Control	logi	yes/no
380	CZ_Display_Status	logi	yes/no
390	CZ_Direction	inte	>9
400	CZ_Arm_State	inte	>9
410	CZ_Disarm_State	inte	>9
420	Unlock_Grace_Time	inte	>>>9
430	Open_Grace_Time	inte	>>>9
440	Close_Debounce_Time	inte	>>>9
450	Lock_Release_Change_Time	inte	>>>9
460	Bond_Sensor_Change_Time	inte	>>>9
470	Latch_Bolt_Change_Time	inte	>>>9
480	Cam_Sensor_Change_Time	inte	>>>9
490	DSM_A_Change_Time	inte	>>>9
500	DSM_B_Change_Time	inte	>>>9
510	RTE_Change_Time	inte	>>>9
520	Shunt_Cancel_Disable	logi	yes/no
530	Relock_Expiration	logi	yes/no

Foreign References

Table	Method	Phrase(s)
Reader	Join - Door_ID	<ul style="list-style-type: none"> door of reader reader of door reader of door where reader.related_object_ID = door.door_id and reader.door_reader_number = PRM\$OST_Reader_1 21 - Reader 1 of a door PRM\$OST_Reader_2 22 - Reader 2 of a door PRM\$OST_Reader_3 74 - Reader 3 (asset 1) PRM\$OST_Reader_4 75 - Reader 4 (asset 2) PRM\$OST_Reader_1_2 76 - Reader 1 and 2 PRM\$OST_Reader_3_4 77 - Reader 3 and 4 PRM\$OST_Reader_1_3 78 - Reader 1 and 3 PRM\$OST_Reader_2_4 79 - Reader 2 and 4 PRM\$OST_Reader_1_2_3_4 80 - Reader 1,2,3,4
Groups	table - Group_Member	<ul style="list-style-type: none"> groups of group_member where group_member.object_id = door.door_id each group_member of groups, each door where door.door_id = group_member.object_id
Maps	table - Map_Icon	<ul style="list-style-type: none"> maps of map_icon where map_icon.object_id = door.door_id each map_icon of maps, each door where door.door_id = map_icon.object_id
Inputs	Table - Inputs	<ul style="list-style-type: none"> inputs.related_object_id = door.door_id inputs.door_relation = {&PRM\$OST_Door_Switch_Monitor} or {&PRM\$OST_Request_To_Exit} or {&PRM\$OST_Lock_Status_Sensor}}
Outputs	Table - Outputs	<ul style="list-style-type: none"> outputs.related_object_id = door.door_id outputs.door_relation = {&PRM\$OST_Unlocked} or {&PRM\$OST_ADA_Unlocked}

Table	Method	Phrase(s)
Event	Table - Actions (door 'accesses' or 'rejects' trigger events)	<ul style="list-style-type: none"> • each event where actions.source_id = door.door_id no-lock no-error • each door where door.door_id = actions.target_id, if available.
	Table - Actions (event activation causes door lock/unlock)	<ul style="list-style-type: none"> • each event where actions.source_id = event.event_id no-lock, • each door where door.door_id = actions.target_id [shared-lock no-wait no-error. if available event / door then . else ...]
Clearance	Table - Clear_Pair	<ul style="list-style-type: none"> • each clear_pair of clearance, each door where door.door_id = clear_pair.object_id • each clearance of clear_pair where clear_pair.object_id = door.door_id

Internal References

Table	Method	Phrase(s)
Unit	Join - Unit_ID	<ul style="list-style-type: none"> • unit of door
Area	Join – Mapping_Number	<ul style="list-style-type: none"> • area of door • door af area
RF_Controller	Join – RF_Controller_ID	<ul style="list-style-type: none"> • rf_controller of door
Asset	Join – Asset_Unit_ID	<ul style="list-style-type: none"> • asset of door • door of control_zone
Control_Zone	Join – Controle_Zone_ID	<ul style="list-style-type: none"> • door of control zone
Handled_Assets	Join – Mapping_Number	<ul style="list-style-type: none"> • handled assets of door
Partition	Join – Partition_ID	<ul style="list-style-type: none"> • door of partition

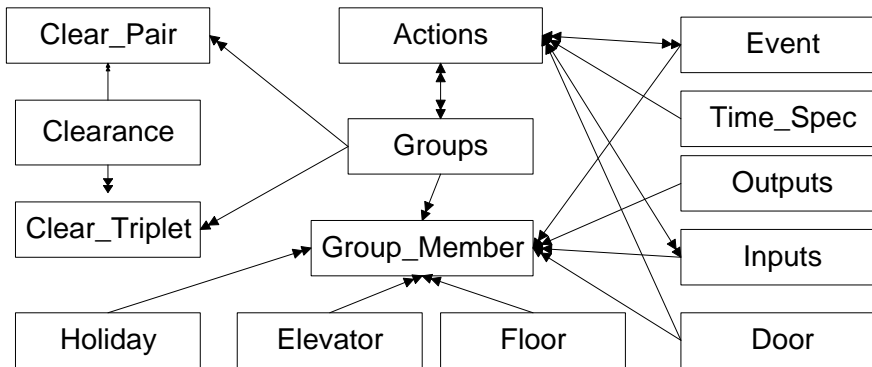
Groups Object

Business Definition

An arbitrary collection of objects of the same type.

Relational Description

- A group consists of group members which may reference one or many Inputs, Outputs, Events, Doors, Elevators, Floors, or Holidays,
- A group may be related to an event through the Actions table (where the group id is the target).
- A group may be related to a clearance through the Clear_Pair or Clear_Triplet tables.
- A group may be related to various objects through the Actions table where the group is the target and the source may be a time_spec, an input, or a door.



Primary Tables

Groups

Index	Fields	Type
Group_ID	Group_ID	primary unique
Group_Name	Group_Name	unique
Object_Type		active

Order	Field Name	Data Type	Flags	Format
10	Group_ID	inte	im	->>>>>>>>>9
20	Object_Type	inte		>>9
30	Group_Name	char	i	X(50)
40	Description	char		X(3000)
50	Partition_ID	inte	im	->>>>>>>>>9
51	Log1	logi		yes/no

Group_Member

Index	Fields	Type
Group_Member	Group_ID Object_ID	primary unique
Object_ID	Object_ID	active

Order	Field Name	Data Type	Flags	Format
10	Group_ID	inte	im	->>>>>>>>>9
20	Object_ID	inte	im	->>>>>>>>>9

Internal References

Table	Method	Phrase(s)
Groups	Join - Group_ID	<ul style="list-style-type: none"> group_member of groups groups of group_member

Foreign References

Table	Method	Phrase(s)
Holidays	Table - Group_Member	(see Holidays)
Inputs	Table - Group_Member	(see Inputs)
Outputs	Table - Group_Member	(see Outputs)
Door	Table - Group_Member	(see Door)
Event	Table - Group_Member	(see Event)
Elevator	Table - Group_Member	(see Elevator)
Floor	Table - Group_Member	(see Floor)
Obj_Archive	Table - Group_Member	Group_Member of Obj_Archive
Group_Member	Join - Group_ID	Group_Member of Groups
Area	Table - Group_Member	
Reader	Table - Group_Member	

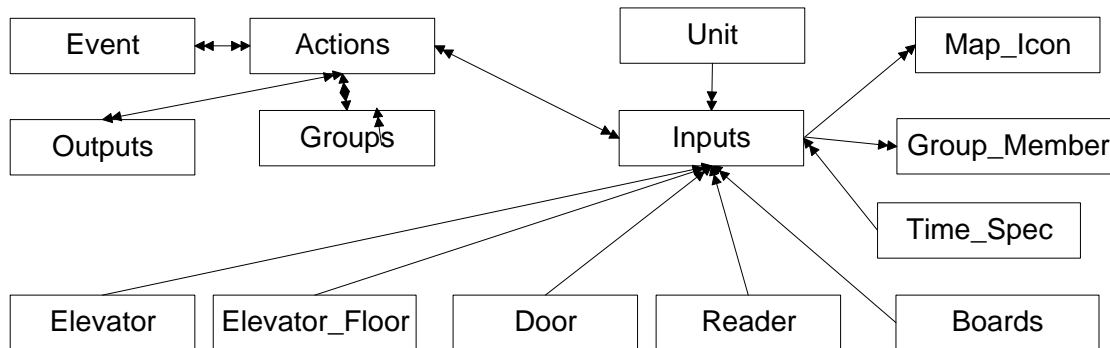
Inputs Object

Business Definition

A device which monitors a switch closure signifying a door opening, a broken window, etc.

Relational Description

- Each input is directly related to a unit and to a board or a reader.
- May belong to a group.
- May be related to one, two, or three events (or event groups) through the Actions table where the input is the source_id.
- May be related to one or two outputs (or output groups) through the Actions table where the input is the source_id.
- May be directly related to a door as the &prm\$ost_door_switch_monitor or &prm\$ost_request_to_exit (using the door_relation field).
- May be the target of one or many events through the Actions table.
- May be related to one or many maps through the map_icon table.
- May be directly related to one time_spec.
- May be directly related to one elevator.
- May be directly related to one or more elevator floors, depending on the input selection mode of the elevator (See Elevator).



Primary Tables

Inputs

Index	Fields	Type
Input_ID	Input_ID	primary unique
Input_Name	Input_Name	unique
Board_ID	Board_ID	active
Unit_ID	Unit_ID	active
Time_spec_ID	Time_spec_ID	active
Door_Index	Door_ID Door_Relation	unique

Order	Field Name	Data Type	Flags	Format
10	Input_ID	inte	im	->>>>>>>>9
20	Unit_ID	inte	i	->>>>>>>>9
30	Board_ID	inte	i	->>>>>>>>9
40	Time_Spec_ID	inte	i	->>>>>>>>9
50	Related_Object_ID	inte	i	->>>>>>>>9
60	Door_Relation	inte	i	>, >>9
70	Online	logi		yes/no
80	Slot_Index	inte		->>>>>>>>9
90	Reverse_Input_Sense	logi		yes/no
100	Activate_on_Supervision	logi		yes/no
110	Annunciate	logi		yes/no
120	Armed	logi		yes/no
130	Input_Name	char	i	X(50)
140	Description	char		X(3000)
150	Control_Zone_ID	inte	i	->, >>>, >>>9
160	BiDirectional_Rad_Alarm_Code	char		X(10)
170	BiDirectional_Rad_Restore_Code	char		X(10)
180	Activate_GP_Protocol_ID	inte		>>>>>>>>9
190	Deactivate_GP_Protocol_ID	inte		>>>>>>>>9
200	Error_GP_Protocol_ID	inte		>>>>>>>>9
210	Error_Restore_GP_Protocol_ID	inte		>>>>>>>>9
220	CZ_Entrance_Delay_Trigger	logi		yes/no
230	CZ_Entrance_Delay_Shunt	logi		yes/no
240	Tamper_GP_Protocol_ID	inte		>>>>>>>>9
250	Tamper_Restore_GP_Protocol_ID	inte		>>>>>>>>9
260	Low_Battery_GP_Protocol_ID	inte		>>>>>>>>9
270	Battery_Restore_GP_Protocol_ID	inte		>>>>>>>>9
280	Timeout_In_Seconds	inte		>>>>>9
290	Board_Type	inte		>>9
300	Send_To_Journal	logi		yes/no

Internal References

Table	Method	Phrase(s)
Unit	Join - Unit_ID	<ul style="list-style-type: none"> inputs of unit
Boards	Join - Board_ID	<ul style="list-style-type: none"> inputs of boards
Door	Join - Door_ID	<ul style="list-style-type: none"> inputs of door

Table	Method	Phrase(s)
Elevator	Join - Elevator_ID	<ul style="list-style-type: none"> elevator of inputs
Elevator_Floor	Join - Elevator_ID	<ul style="list-style-type: none"> elevator_floor of inputs
Time_Spec	Join - Time_Spec_ID	<ul style="list-style-type: none"> inputs of time_spec
GP_Receiver	Join - Unit_ID	<ul style="list-style-type: none"> inputs of GP_Receiver
GP_Gateway	Join - Unit_ID	<ul style="list-style-type: none"> inputs of GP_Gateway

External References

Table	Method	Phrase(s)
Reader	exceptional code	<ul style="list-style-type: none"> find reader where reader.reader_id = inputs.board_id find inputs where inputs.board_id = reader.reader_id time_spec of inputs
Groups	Table - Group_Member (Input Group)	<ul style="list-style-type: none"> groups of group_member where group_member.object_id = inputs.input_id each group_member of groups, each event where inputs.input_id = group_member.object_id
	Table - Actions (Event Group) (Input is the source; inputs activate or arm event groups)	<ul style="list-style-type: none"> actions.source_id = inputs.inputs_id, each groups where groups.group_id = actions.target_id and groups.object_type = &prm\$obj_event_group comment: it may be more reliable to use explicit find statements with appropriate lock and available clauses
	Table - Actions (Output Group) (Input is the source; inputs activate output groups)	<ul style="list-style-type: none"> actions.source_id = inputs.inputs_id, each groups where groups.group_id = actions.target_id and groups.object_type = &prm\$obj_event_group comment: it may be more reliable to use explicit find statements with appropriate lock and available clauses
Event	Table - Actions (Input is armed by an event through the Actions table)	<ul style="list-style-type: none"> each actions where actions.source_id = event.event_id no-lock, each inputs where inputs.input_id = actions.target_id [shared-lock no-wait no-error. if available inputs then. else ...]
	Table - Actions (Input is the source; inputs activate events)	<ul style="list-style-type: none"> Refer to Cross-Reference Section.
Outputs	Table - Actions (Input is the source; inputs activate outputs)	<ul style="list-style-type: none"> Refer to Cross-Reference Section.
Maps	Table - Map_Icon	<ul style="list-style-type: none"> each maps of map_icon where map_icon.object_id = inputs.input_id each inputs where inputs.input_id = map_icon.object_id

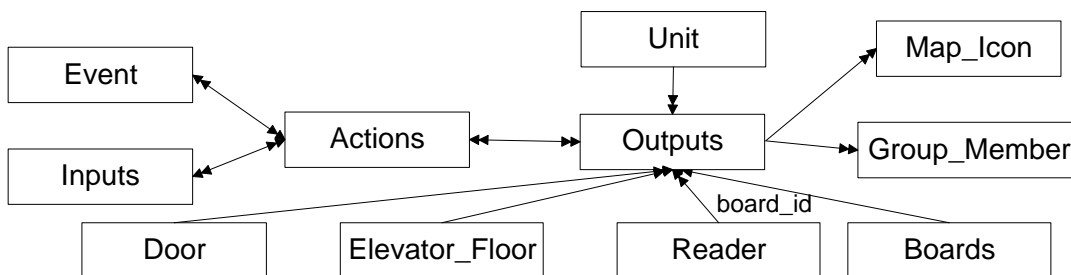
Outputs Object

Business Definition

A connection point on an apC or iSTAR board or on a reader which generates an electrical signal when activated.

Relational Description

- Is directly related to a board or a reader.
- Is directly related to a unit.
- May belong to a group.
- May be the target of an event through the Actions table
- May be related to one or many inputs through the actions table where the output is the target.
- May be directly related to a door when the door relation is &prm\$ost_unlocked or &prm\$ost_ada_unlocked.
- May be related to one or many maps through the map_icon table.
- May be related to one elevator through the elevator_floor table
- May be related to one floor through the elevator_floor table



Primary Tables

Outputs

Index	Fields	Type
Output_ID	Output_ID	primary unique
Output_Name	Output_Name	unique
Board_ID	Board_ID	active
Unit_ID	Unit_ID	active
Door_Index	Related_Object_ID Door_Relation	unique

Order	Field Name	Data Type	Flags	Format
10	Output_ID	inte	im	->>>>>>>>>9
20	Unit_ID	inte	i	->>>>>>>>>9
30	Board_ID	inte	i	->>>>>>>>>9
40	Related_Object_ID	inte	i	->>>>>>>>>9
50	Door_Relation	inte	i	>, >>9
60	Online	logi		yes/no
70	Slot_Index	inte		>>9
80	Normally_Energized	logi		yes/no
90	Pulse_Duration	inte		>>9
100	Output_Name	char	i	X(50)
110	Description	char		X(3000)
120	Annunciate	logi		yes/no
130	Board_Type	inte		>>9
140	Send_To_Journal	logi		yes/no

Internal References

Table	Method	Phrase(s)
Unit	Join - Unit_ID	<ul style="list-style-type: none"> outputs of unit
Boards	Join - Board_ID	<ul style="list-style-type: none"> outputs of board board of outputs
Elevator_Floor	Join - Output_ID	<ul style="list-style-type: none"> outputs of elevator_floor elevator_floor of outputs

External References

Table	Method	Phrase(s)
Reader	exceptional code	<ul style="list-style-type: none"> find reader where reader.reader_id = outputs.board_id find outputs where outputs.board_id = reader.reader_id
Door	exceptional code	<ul style="list-style-type: none"> find door where door.door_id = outputs.related_object_id find outputs where outputs.related_object_id = door.door_id
Elevator	Table - Elevator_Floor	<ul style="list-style-type: none"> elevator_floor of outputs; elevator of elevator_floor
Groups	Table - Group_Member (Output Group)	<ul style="list-style-type: none"> groups of group_member where group_member.object_id = outputs.output_id each group_member of Groups, each event where outputs.output_id = group_member.object_id

Event,
Inputs

Table - Actions

- each actions where actions.source_id = event.event_id ; where actions.target_id = outputs.output_id
- each actions where actions.source_id = inputs.input_id.

Maps

Table - Map_Icon

- maps of map_icon where map_icon.object_id = outputs.output_id
- each map_icon of maps, each output where output.output_id = map_icon.object_id

Person Object

Business Definition

An individual user of the system; may be a system operator or an individual who uses their access credentials for access or both. Tables also support:

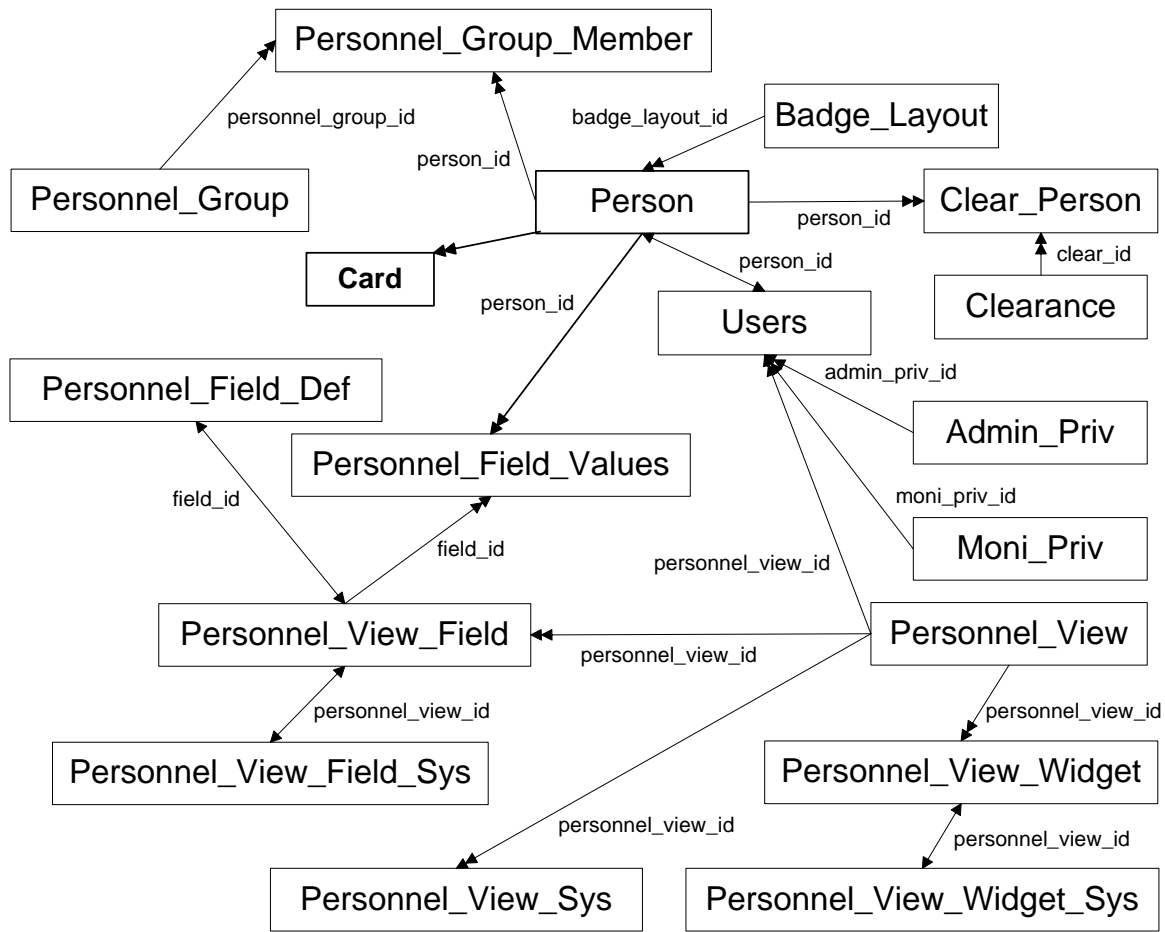
- Information for defining a system User, someone who can log into the Administration and/or Monitoring stations.
- Identification of Administration and Monitoring Privileges for each User.
- The definition and storage of user defined fields.
- The creation of enumeration lists that define the legal values for specified fields.
- The screen layouts of personnel views.
- The definition of personnel groups, which are used for asset tracking

Relational Description

A person:

- May be directly related to one badge layout.
- May be related to one or more access credentials through the card table.*
- May be directly related to one user (ie: system operator of ccure800).
- May be related to one or many clearances through the clear_person table.
- May be related to one or more personnel_group through the personnel_group_member table.
- May be related to one admin_priv (admin privilege) through the users table.
- May be related to one moni_priv (monitoring privilege) through the users table.
- May be related to one personnel_view (personnel view) through the users table.

* **NOTE:** In version 9.1 of C•Cure 800/8000, a person can relate to multiple cards.



Primary Tables

Person

Index	Fields	Type
CardNum_Del_IDX	Card_num Deleted	Primary Active
Deleted	Deleted	Active
First_Name	First_Name	Active
Int1	Int1	active
Int2	Int2	active
Int6	Int6	Active
Int7	Int7	Active unique
Last_Mod_DT	Last_Mod_DT	active
Last_Name_IDX	Last_Name First_Name	active
Partition_ID	Partition_ID	active
Person_GID	Person_GID	active
Person_ID	Person_ID	unique
Text1	Text1	active
Text2	Text2	active
Text6	Text6	Active unique
Text7	Text7	Active unique
Text8	Text8	active

Index	Fields	Type
Text10	Text10	active
Update_Count	Update_Count	active
Person_EUID	Person_EUID	Active unique

Order	Field Name	Data Type	Flags	Format
10	Person_ID	inte	im	->>>>>>>>>9
20	First_name	char	i	x(63)
30	Middle_Name	char		x(60)
40	Last_Name	char	i	X(63)
50	Person_Type	inte		99
60	Card_num	deci	i	>>>>>>>>>9
70	PIN	inte		>>>>>>>>>9
80	Badge_Layout_ID	inte		->>>>>>>>>9
130	Issue_Code	inte		>>>>>>>>>9
140	Facility_code	inte		>>>>>>>>>9
150	Inactive	logi		yes/no
160	Expired	logi		yes/no
170	Disabled	logi		yes/no
180	ADA	logi		yes/no
190	Deleted	logi	i	yes/no
210	Lost	logi		yes/no
220	Noticed	logi		yes/no
230	Last_Mod_DT	inte	i	->, >>>, >>>, >>>9
240	Last_Mod_Person	inte		->>>>>>>>>9
250	Text1	char	i	X(255)
260	Text2	char	i	X(255)
270	Text3	char		X(255)
280	Text4	char		X(255)
290	Text5	char		X(255)
300	Int1	inte	i	->>>>>>>>>9
310	Int2	inte	i	->>>>>>>>>9
320	Int3	inte		->>>>>>>>>9
330	Int4	inte		->>>>>>>>>9
340	Int5	inte		->>>>>>>>>9
350	Logical1	logi		yes/no
360	Logical2	logi		yes/no
370	Date1	date		99/99/9999
380	Date2	date		99/99/9999
390	Is_User	logi		yes/no
400	Activate_AP_Event	logi		yes/no
420	Asset_Administrator	logi		yes/no
450	Has_been_user	logi	m	yes/no
460	Image_Capture_DT	inte		>>>>>>>>>9
470	Badge_Print_DT	inte		>>>>>>>>>9
480	Signature_Capture_DT	inte		>>>>>>>>>9

Flags	Index Name	Field Name
	Card_ID	Card_ID
p	Card_Idx	Person_ID + Card_num + Deleted
	Card_Num	Card_num
	CHUID	CHUID
	Government_Idx	Agency_Code + System_Code + Card_num + Credential_Series + Credential_Issue
	Is_Extended	Is_Extended
	Is_Primary	Is_Primary
	Person_Id	Person_ID
	Person_Idx	Person_ID + Card_ID
	PIN_Access	PIN_Access

CHUID_Format

Purpose:

- Reserved for storing CHUID (Cardholder Unique Identifier) Format that applies to this particular installation.

Index	Fields	Type

Table: CHUID_Format

Order	Field Name	Data Type	Flags	Format
10	CHUID_Format_ID	inte	im	->, >>>, >>9
20	CHUID_Format_Name	char	m	x(50)
30	Description	char		X(3000)
40	Update_ID	inte		->>>>>>>>9
50	CHUID_Length	inte		>>>>>>>>9

Table: CHUID_Format_Field

Order	Field Name	Data Type	Flags	Format
10	Field_ID	inte	im	->>>>>>>>9
20	CHUID_Format_ID	inte	im	->, >>>, >>9
30	Field_Position	inte		>>>>>9
40	Field_Length	inte		>>>>>9
50	Field_Order	inte		>>>>>9
60	Field_Label	char		X(50)

Badge_Layout

Index	Fields	Type
Badge_Layout_ID	Badge_Layout_ID	primary unique
Badge_Layout_Name	Badge_Layout_Name	unique
Badge_Layout_EUID	Badge_Layout_EUID	Unique active

Order	Field Name	Data Type	Flags	Format
10	Badge_Layout_ID	inte	im	->>>>>>>>9
20	Badge_Layout_Name	char	i	X(50)
30	Description	char		X(3000)
40	Filespec	char		X(320)
50	HasSignature	logi		yes/no
60	Stripe_Format_1	inte		->, >>>, >>9
70	Stripe_Format_2	inte		->, >>>, >>9
80	Stripe_Format_3	inte		->, >>>, >>9
90	Layout_file_edit_DT	inte		>>>>>>>>9
100	Badge_Layout_EUID	char	i	X(40)

Foreign References

Table	Method	Phrase(s)
Person	Join - Badge_Layout_ID	<ul style="list-style-type: none"> • badge_layout of person • person of badge_layout

Users

Index	Fields	Type
Person_ID	Person_ID	primary unique
User_Name	User_Name	unique
Admin_Priv_ID	Admin_Priv_ID	active
Moni_Priv_ID	Moni_Priv_ID	active
Personnel_View_ID	Personnel_View_ID	active

Order	Field Name	Data Type	Flags	Format
10	Person_ID	inte	im	->>>>>>>>9
20	Password	char		x(20)
30	Default_Screen_ID	inte		->>>>>>>>9
60	Enable_Breakthrough	logi		yes/no
80	Enabled	logi		yes/no
90	User_Name	char	i	x(20)
120	Admin_Priv_ID	inte	i	->>>>>>>>9
130	Moni_Priv_ID	inte	i	->>>>>>>>9
140	Personnel_View_ID	inte	i	->>>>>>>>9
141	User_type	inte	m	>9
151	ODBC_Enabled	logi		yes/no
500	Is_Partition_Admin	logi		yes/no
510	TOD_Time_Spec_ID	inte		->>>>>>>>9
511	TOD_Moni_Priv_ID	inte		->>>>>>>>9
512	TOD_Event_ID	inte		->>>>>>>>9
513	TOD_Moni_Priv_Enabled	logi		yes/no
514	TOD_Not_Trigger_Last_Event	logi		yes/no
515	ODBCPwd_LastModDT	inte		->,>>>,>>>,>>9
516	Is_Limited_Partition_Admin	logi		yes/no

Table: Partition

Order	Field Name	Data Type	Flags	Format
10	Partition_ID	inte	im	->>>>>>>>9
20	Partition_Name	char	im	X(50)
30	Description	char		X(3000)
40	Is_Shared	logi		yes/no
50	Partition_EUID	char	i	X(40)
60	Is_Shared_Limited	logi		yes/no

Table: Partition_Map

Order	Field Name	Data Type	Flags	Format
10	Target_ID	inte	im	->>>>>>>>9
11	Base_ID	inte	im	->>>>>>>>9
12	Partition_ID	inte	im	->>>>>>>>9

Table: Partition_Share_Limits

Order	Field Name	Data Type	Flags	Format
10	Shared_Partition_ID	inte	im	->>>>>>>>9
20	Shared_With_Partition_ID	inte	im	->>>>>>>>9

Admin_Priv

Index	Fields	Type
Admin_Priv_ID	Admin_Priv_ID	primary unique
Admin_Priv_Name	Admin_Priv_Name	unique

Order	Field Name	Data Type	Flags	Format
10	Admin_Priv_ID	inte	im	->>>>>>>>9
20	Admin_Priv_Name	char	i	X(20)
30	Enabled	logi		yes/no
40	Description	char		X(3000)
50	Version	inte		>,>>>,>>9

Admin_Priv_Screen

Index	Fields	Type
Admin_Priv_Order	Admin_Priv_ID Access_Level Menu_Order Screen_Num	primary unique
Menu_Order	Menu_Order	active
Priv_ID_Screen	Admin_Priv_ID Screen_Num	active
Screen_Num	Screen_Num	active

Order	Field Name	Data Type	Flags	Format
10	Admin_Priv_ID	inte	im	->>>>>>>>>9
20	Screen_Num	inte	im	>>9
30	Menu_Order	inte	im	>9
40	Access_Level	inte	im	9

Moni_Priv

Index	Fields	Type
Moni_Priv_ID	Moni_Priv_ID	primary unique
Moni_Priv_Name	Moni_Priv_Name	unique

Order	Field Name	Data Type	Flags	Format
10	Moni_Priv_ID	inte	im	->>>>>>>>>9
20	Moni_Priv_Name	char	i	X(50)
30	Enabled	logi		yes/no
40	Description	char		X(3000)
50	Show_Deleted	logi	m	yes/no
60	Run_Moni_Station	logi	m	yes/no
70	Default_Priority	inte	m	>>9
80	Max_Priority	inte	m	>>9
90	Allow_Gracing_Cards	logi		yes/no
100	Set_Server_Connections	logi		yes/no
110	Image_File_Dir_Enabled	logi		yes/no
120	Disable_Man_Act_Challenge	logi		yes/no

Moni_Priv_Msg

Index	Fields	Type
Priv_ID_Msg_Num	Moni_Priv_ID Message_Num	primary unique
Msg_Num	Message_Num	

Order	Field Name	Data Type	Flags	Format
10	Moni_Priv_ID	inte	im	->>>>>>>>9
20	Message_Num	inte	im	>>9

Moni_Priv_Object

Index	Fields	Type
Moni_Priv_Object	Moni_Priv_ID Moni_Object_Type Object_ID	primary unique
Object_ID	Object_ID	

Order	Field Name	Data Type	Flags	Format
10	Moni_Priv_ID	inte	im	->>>>>>>>9
20	Object_ID	inte	im	->>>>>>>>9
30	Moni_Object_Type	inte	im	>>9
40	Manual_Action_Allowed1	inte		->, >>>, >>>, >>9
50	Manual_Action_Allowed2	inte		->, >>>, >>>, >>9
60	Manual_Action_Allowed3	inte		->, >>>, >>>, >>9
70	Manual_Action_Allowed4	inte		->, >>>, >>>, >>9
80	Manual_Action_Allowed5	inte		->, >>>, >>>, >>9
90	Manual_Action_Allowed6	inte		->, >>>, >>>, >>9
100	Manual_Action_Allowed7	inte		->, >>>, >>>, >>9

Personnel_View

Index	Fields	Type
Personnel_View_Unique	Personnel_View_ID Language Screen_Type	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
30	Description	char		X(3000)
40	Num_Tab_Pages	inte		->, >>>, >>9
60	Tab_Pages_Y_Pos	inte		->, >>>, >>9
70	Tab_Pages_X_Pos	inte		->, >>>, >>9
80	Tab_Pages_Y_Size	inte		->, >>>, >>9
90	Tab_Pages_X_Size	inte		->, >>>, >>9
100	View_X_Size	inte		->, >>>, >>9
110	View_Y_Size	inte		->, >>>, >>9
120	Tab_Page_Names	char		X(3000)
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_View_Field

Index	Fields	Type
Personnel_View_Field	Personnel_View_ID Language Screen_Type Field_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
20	Field_ID	inte	im	->>>>>>>>9
30	Page_Number	inte		->, >>>, >>9
50	X_Pos	inte		->, >>>, >>9
60	Y_Pos	inte		->, >>>, >>9
70	X_Size	inte		->, >>>, >>9
80	Y_Size	inte		->, >>>, >>9
90	Read_Only	logi		yes/no
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_View_Field_Sys

Index	Fields	Type
Personnel_View_Field_Sys	Personnel_View_ID Language Screen_Type Field_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
20	Field_ID	inte	im	->>>>>>>>9
30	Page_Number	inte		->, >>>, >>9
50	X_Pos	inte		->, >>>, >>9
60	Y_Pos	inte		->, >>>, >>9
70	X_Size	inte		->, >>>, >>9
80	Y_Size	inte		->, >>>, >>9
90	Read_Only	logi		yes/no
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_View_Sys

Index	Fields	Type
Personnel_View_Field_Sys	Personnel_View_ID Language Screen_Type	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
30	Description	char		X(3000)
40	Num_Tab_Pages	inte		->, >>>, >>9
60	Tab_Pages_Y_Pos	inte		->, >>>, >>9
70	Tab_Pages_X_Pos	inte		->, >>>, >>9
80	Tab_Pages_Y_Size	inte		->, >>>, >>9
90	Tab_Pages_X_Size	inte		->, >>>, >>9
100	View_X_Size	inte		->, >>>, >>9
110	View_Y_Size	inte		->, >>>, >>9
120	Tab_Page_Names	char		X(3000)
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_View_Widget

Index	Fields	Type
Personnel_View_Widget	Personnel_View_ID Language Screen_Type Widget_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
20	Widget_ID	inte	im	->>>>>>>>9
30	Widget_Type	inte		->, >>>, >>9
40	Widget_Label	char		X(256)
50	Page_Number	inte		->, >>>, >>9
70	X_Pos	inte		->, >>>, >>9
80	Y_Pos	inte		->, >>>, >>9
90	X_Size	inte		->, >>>, >>9
100	Y_Size	inte		->, >>>, >>9
110	Disabled	logi		yes/no
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_View_Widget_Sys

Index	Fields	Type
Personnel_View_Widget_Sys	Personnel_View_ID Language Screen_Type Widget_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_View_ID	inte	im	->>>>>>>>9
20	Widget_ID	inte	im	->>>>>>>>9
30	Widget_Type	inte		->, >>>, >>9
40	Widget_Label	char		X(256)
50	Page_Number	inte		->, >>>, >>9
70	X_Pos	inte		->, >>>, >>9
80	Y_Pos	inte		->, >>>, >>9
90	X_Size	inte		->, >>>, >>9
100	Y_Size	inte		->, >>>, >>9
110	Disabled	logi		yes/no
130	Language	inte	i	->>>>>>>>9
140	Screen_Type	inte	i	->>>>>>>>9

Personnel_Field_Def

Index	Fields	Type
Field_ID	Field_ID Deleted	primary unique

Order	Field Name	Data Type	Flags	Format
20	Field_ID	inte	im	->>>>>>>>9
40	Type	inte		>>9
60	Mandatory	logi		yes/no
70	Enumerated	logi		yes/no
80	Num_Values	inte		>>9
90	Max_Length	inte		>>9
100	Min_Value	char		x(20)
110	Max_Value	char		x(20)
120	DefOutLength	inte		>>9
130	MinOutLength	inte		>>9
140	Enum_List_Last_SeqID	inte		->, >>>, >>9
150	Deleted	logi	i	yes/no
160	IgnoreHomeNode	logi		yes/no
170	HomeServerEuid	char		X(40)
180	GlobalMappingEuid	char	i	X(40)
190	GlobalFieldName	char	i	X(50)
200	isGlobal	logi		yes/no
210	EditingLevel	inte		>>9

Personnel_Field_Values

Index	Fields	Type
Field_Chr_Idx	Field_ID Char_Value Pers_Field_SeqID	primary unique
Field_Dec_Idx	Field_ID Decimal_Value Pers_Field_SeqID	unique
Person_Field	Person_ID Field_ID	unique
Pers_F_Idx	Pers_Field_SeqID	unique

Order	Field Name	Data Type	Flags	Format
10	Person_ID	inte	im	->>>>>>>>9
20	Field_ID	inte	im	->>>>>>>>9
30	Char_Value	char	i	X(30)
40	Decimal_Value	deci-2	i	->>, >>9.99
50	Pers_Field_SeqID	inte	i	->, >>>, >>9

Personnel_Group

Index	Fields	Type
Personnel_Group_ID	Personnel_Group_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Personnel_Group_ID	inte	im	->>>>>>>>9
20	Personnel_Group_Name	char		X(50)
30	Description	char		X(3000)
40	Partition_ID	inte	im	->>>>>>>>9
50	Is_Carpool	logi		yes/no

Personnel_Group_Member

Index	Fields	Type
Group_Member	Personnel_Group_ID Person_ID	primary unique
Person_ID	Person_ID	primary

Order	Field Name	Data Type	Flags	Format
10	Personnel_Group_ID	inte	im	->>>>>>>>9
20	Person_ID	inte	im	->>>>>>>>9
30	Slot	inte		>,>>>,>>9
40	Is_Carpool	logi		yes/no

Foreign References

Table	Method	Phrase(s)
Badge_Layout	Join - Badge_Layout_ID	<ul style="list-style-type: none"> person of badge_layout badge_layout of person
Clearance	Table - Clear_Person	<ul style="list-style-type: none"> each clearance of clear_person, each clear_person of person each person of clear_person, each clear_person of clearance
Admin_Priv	Join - Admin_Priv_ID	<ul style="list-style-type: none"> admin_priv of user each user of admin_priv
Moni_Priv	Join - Moni_Priv_ID	<ul style="list-style-type: none"> moni_priv of user each user of moni_priv
Personnel_View	Join - Personnel_View_ID	<ul style="list-style-type: none"> personnel_view of user each user of personnel_view

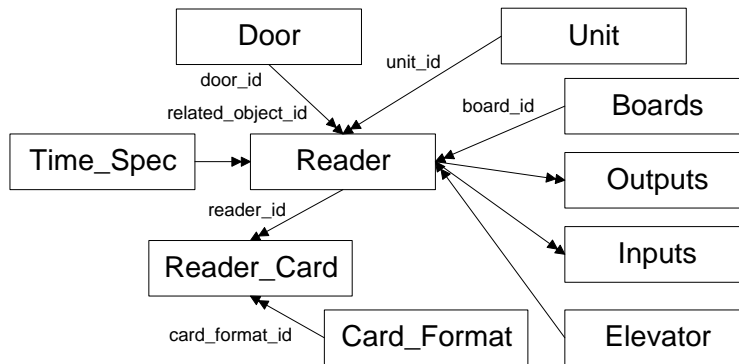
Reader Object

Business Definition

A device capable of reading the information of an access control card and passing this information to an apC.

Relational Description

- Is directly related to a board and a unit.
- May be directly related to one or two inputs where the input board_id is the reader_id.
- May be directly related to one or two outputs where the output board_id is the reader_id.
- May be directly related to a door as reader 1, reader 2, reader 3 (asset), or reader 4 (asset).
- May be directly related to one elevator via the elevator reader_id (in this case the related_object_id of the reader is the object id of the elevator).



Primary Tables

Reader

Index	Fields	Type
Reader_ID	Reader_ID	primary unique
Reader_Name	Reader_Name	unique
Board_ID	Board_ID	active
Unit_ID	Unit_ID	active
Related_Object_ID	Related_Object_ID	active

Order	Field Name	Data Type	Flags	Format
10	Reader_ID	inte	im	->>>>>>>>9
20	Unit_ID	inte	i	->>>>>>>>9
30	Board_ID	inte	i	->>>>>>>>9
40	Related_Object_ID	inte	i	->>>>>>>>9
50	Door_Reader_Number	inte		>>9
60	Online	logi		yes/no
70	LCD	logi		yes/no

80	Keypad	logi		yes/no
90	Reader_Status	inte		>9
100	Technology	inte		>9
110	Type	inte		>9
120	Slot_Index	inte		>>9
130	Reader_Name	char	i	X(50)
140	Description	char		X(3000)
150	PIN_Time_Spec_Id	inte		->>>>>>>>>9
160	Allow_Keypad_Card_Number	logi		yes/no
170	Annunciate	logi		yes/no
180	Always_Require_Pin	logi		yes/no
190	ipx_Is_Primary	logi		yes/no
200	Cards_Always_Rejected	logi		yes/no
210	Keypad_Com_Permission	inte		>9
220	Keypad_Com_TimeSpec_ID	inte		->>>>>>>>>9
230	Status_Msg_ID	inte		->>>>>>>>>9
240	Paired_Reader_ID	inte		->>>>>>>>>9
250	Board_Type	inte		>>9
260	Allow_Pin_Only_Access	logi		yes/no

Internal References

Table	Method	Phrase(s)
Unit	Join - Unit_ID	<ul style="list-style-type: none"> • reader of unit
Boards	Join - Board_ID	<ul style="list-style-type: none"> • reader of boards
Reader_Card	Join - Reader_ID	<ul style="list-style-type: none"> • reader_card of reader
Sequence	Join - Type	<ul style="list-style-type: none"> • reader of sequence
Elevator	Join - Reader_ID	<ul style="list-style-type: none"> • reader of elevator • elevator of reader
Time_Spec	Join - Time_Spec_ID	<ul style="list-style-type: none"> • time_spec of reader

Foreign References

Table	Method	Phrase(s)
Inputs	exceptional code	<ul style="list-style-type: none"> • find reader where reader.reader_id = inputs.board_id • find inputs where inputs.board_id = reader.reader_id
Outputs	exceptional code	<ul style="list-style-type: none"> • find reader where reader.reader_id = outputs.board_id • find outputs where outputs.board_id = reader.reader_id
Door	exceptional code	<ul style="list-style-type: none"> • find door where door.door_id = reader.related_object_id (note, this object may be an elevator) • find reader where reader.related_object_id = door.door_id
Elevator	exceptional code	<ul style="list-style-type: none"> • find elevator where elevator.reader_id = reader.related_object_id • find reader where reader.related_object_id = elevator.reader_id

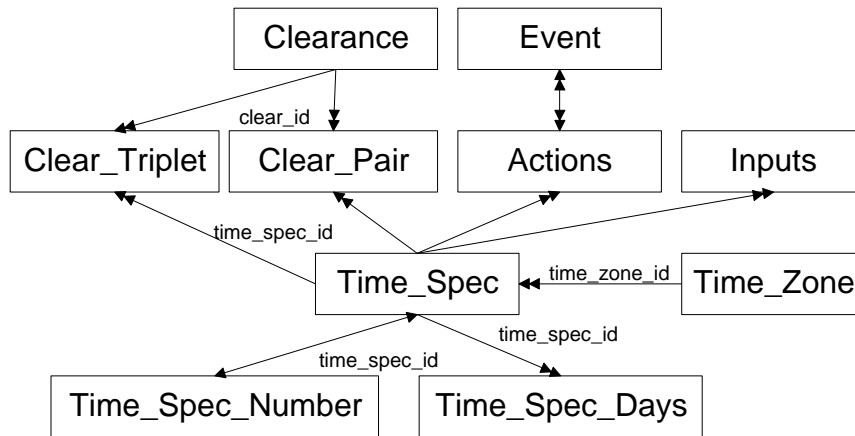
Time Spec Object

Business Definition

A time interval which has particular significance from an access control perspective .

Relational Description

- Consists of time spec days records.
- Is related to one time_spec_number.
- May be related to one or two events through the Actions table where Time_Spec is the source_id.
- May be related to one or many clearances through the clear_pair table.
- May be directly related to one or many inputs.



Primary Tables

Time_Spec

Index	Fields	Type
Time_Spec_ID	Time_Spec_ID	primary unique
Time_Spec_Name	Time_Spec_Name	unique

Order	Field Name	Data Type	Flags	Format
10	Time_Spec_ID	inte	im	->>>>>>>>9
20	Time_Spec_Name	char	i	X(50)
30	Description	char		X(3000)
40	Time_Zone_ID	inte	m	->>>>>>>>9
50	Partition_ID	inte	im	->, >>>, >>9

Time_Spec_Days

Index	Fields	Type
Time_spec_days_index	Time_Spec_ID Day_of_Week Start_Time End_Time	primary unique
Time_Spec_ID	Time_Spec_ID	active

Order	Field Name	Data Type	Flags	Format
10	Time_Spec_ID	inte	im	->>>>>>>>9
20	Day_of_Week	inte	i	9
30	Start_Time	inte	i	->, >>>, >>>, >>9
40	End_Time	inte	i	->, >>>, >>>, >>9

Time_Spec_Number

Index	Fields	Type
Timespec_Number_Index	Time_Spec_ID Time_Spec_Number	primary unique
Deletion_DT	Deletion_DT	active

Order	Field Name	Data Type	Flags	Format
10	Time_Spec_ID	inte	i	->>>>>>>>9
20	Time_Spec_number	inte	im	>>9
30	Deletion_DT	inte	i	->, >>>, >>>, >>9

Internal References

Table	Method	Phrase(s)
Time_Spec	Join - Time_Spec_ID	<ul style="list-style-type: none"> time_spec_days of time_spec time_spec_number of time_spec clear_pair of time_spec clear_triplet of time_spec inputs of time_spec
Time_Zone	Join - Time_Zone_ID	<ul style="list-style-type: none"> time_zone of time_spec

Foreign References

Table	Method	Phrase(s)
Clearance	Clear_Pair Table	<ul style="list-style-type: none"> each time_spec of clear_pair, each clear_pair of clearance each clearance of clear_pair, each clear_pair of time_spec
Elevator	Clear_Triplet Table	<ul style="list-style-type: none"> each clear_triplet of time_spec, find elevator where elevator_or_grp_id = elevator_id
Floor	Clear_Triplet Table	<ul style="list-style-type: none"> each clear_triplet of time_spec, find floor where floor_or_grp_id = floor.floor_id
Group (elevator)	Clear_Triplet Table	<ul style="list-style-type: none"> each clear_triplet of time_spec, find groups where clear_triplet.elevator_or_grp_id = groups.group_id
Groups (floor)	Clear_Triplet Table	<ul style="list-style-type: none"> each clear_triplet of time_spec, find groups where groups.group_id = clear_triplet.floor_or_grp_id
Event	Table - Actions	<ul style="list-style-type: none"> Refer to Cross-Reference Section./

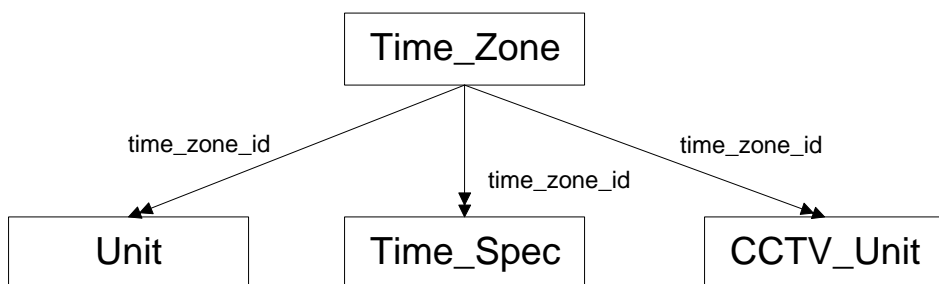
Time Zone Object

Business Definition

A world time zone, such as Eastern Standard Time, Pacific Standard Time, or Greenwich Mean Time.

Relational Description

- May be related to one or more Unit objects through Unit.Time_Zone_ID.
- May be related to one or more Time Spec objects through Time_Spec.Time_Zone_ID.
- May be related to one or more CCTV unit objects through CCTV_Unit.Time_Zone_ID.



Primary Tables

Time_Zone

Index	Fields	Type
Time_Zone_ID	Time_Zone_ID	primary unique
Time_Zone_Name	Time_Zone_Name	unique

Order	Field Name	Data Type	Flags	Format
10	Time_Zone_ID	inte	im	->>>>>>>>>9
20	Time_Zone_Name	char	im	X(50)
30	Description	char		X(3000)
40	Registry_Name	char		X(80)
50	GMT_Offset	inte		->, >>>, >>9
60	Adjust_For_DST	logi		yes/no
70	DST_Offset	inte		->, >>>, >>9
80	DST_Begins_Month	inte		>9
90	DST_Ends_Month	inte		>9
100	DST_Begins_Weekday	inte		>9
110	DST_Ends_Weekday	inte		>9
120	DST_Begins_Week	inte		9
130	DST_Ends_Week	inte		9
140	DST_Begins_Time	inte		>, >>>, >>9
150	DST_Ends_Time	inte		>, >>>, >>9

Internal References

Table	Method	Phrase(s)
Time_Zone	Join - Time_Zone_ID	<ul style="list-style-type: none">• Time_Zone of Unit• Time_Zone of Time_Spec• Time_Zone of CCTV_Unit

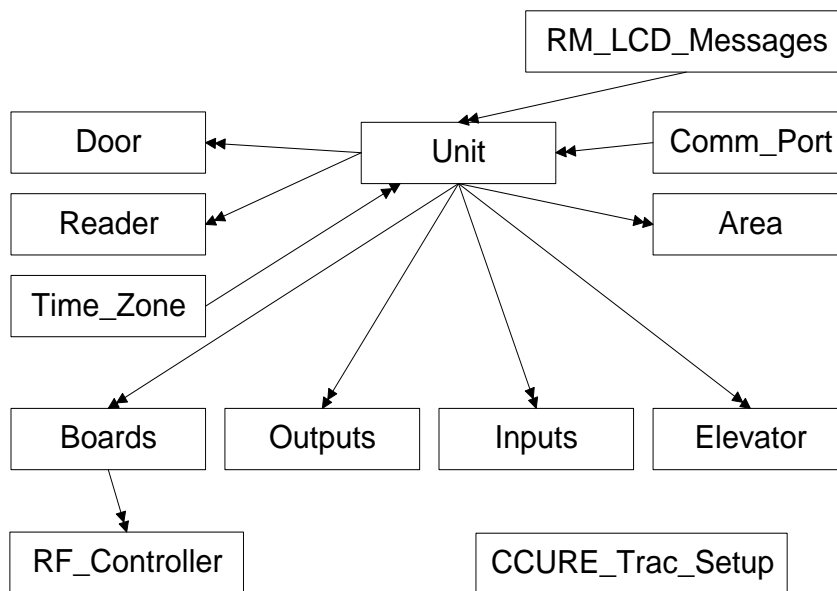
Unit Object

Business Definition

A device which monitors input devices and triggers output devices based on those inputs or commands issued from a guard station. Also known as an apC.

Relational Description

- An apC consists of a unit and one or several boards.
- Is directly related to one comm_port.
- Is directly related to one time_spec.
- Is directly related to one time_zone.
- Is directly related to one rm_lcd_messages.
- May be related to one or several doors, readers, inputs, outputs, elevators, or areas.



Primary Tables

Unit

	<u>Index</u>	<u>Fields</u>	<u>Type</u>		
	Unit_ID	Unit_ID	primary unique		
	Unit_Name	Unit_Name	unique		

Order	Field Name	Data Type	Flags	Format
10	Unit_ID	inte	im	->>>>>>>>9
20	Port_ID	inte		->>>>>>>>9
30	Address	inte		999
40	APC_Type	inte		>>9
50	Unit_Name	char	i	X(50)
60	Description	char		X(3000)
70	Display_Access_On_Open	logi		yes/no
80	Online	logi		yes/no
90	Dialup	logi		yes/no
100	Poll_Period	inte		>>9
110	Wait_Time	inte		>>9
120	Comm_Fail_Delay	inte		>>9
130	Num_Cards	inte		>>>, >>9
140	Percent_Full	inte		>>9
150	Cards_Processed	inte		>>9
160	Comm_Fail_Poll_Period	inte		>>9
170	Comm_Status	inte		>9
180	Download_Status	inte		>9
190	Comm_Fail_Retransmits	inte		>>9
200	Connect_Type	inte	m	9
210	Host_Comm_Fail_Delay	inte		>>9
220	Password	char	m	X(16)
230	Auto_Download_Panel	logi		yes/no
240	Download_Wait	inte		>>>9
250	Times_To_Retry_failed_Conn_Out	inte		>9
260	Comm_Fail_Dial_Interval	inte		>>>9
270	Time_Spec_ID	inte	m	->>>>>>>>9
280	Time_Spec_Dial_Interval	inte		>>>9
290	Outside_Time_Spec_Dial_Interval	inte		>>>9
300	Time_To_Retry_Failed_Conn_In	inte		>9
310	Rmt_Modem_Init_String	char	c	X(20)
320	Rmt_Modem_Resp_Wait_Time	inte		>>9
330	Conn_Host_Wait_Time	inte		>>9
340	Redial_Reconn_Delay_Time	inte		>>9
350	DTR_Off_Time	inte		>>>9
360	Rmt_Modem_Answer_Delay_Time	inte		>>9
370	Rmt_Modem_Status_Check_Time	inte		>>>9
380	Time_Zone_ID	inte	m	->>>>>>>>9
390	RM_LCD_Messages_ID	inte		->>>>>>>>9
400	Percent_Nearly_Full_Event	inte		>>9

Relations:

Reader OF Unit (Unit_ID)
 Inputs OF Unit (Unit_ID)
 Door OF Unit (Unit_ID)
 Outputs OF Unit (Unit_ID)
 Unit_Status OF Unit (Unit_ID)
 Area OF Unit (Unit_ID)
 Elevator OF Unit (Unit_ID)
 Remote_Phone OF Unit (Unit_ID)
 Dial_In OF Unit (Unit_ID)
 Dial_Out OF Unit (Unit_ID)
 Control_Zone OF Unit (Unit_ID)
 Unit OF Time_Spec (Time_Spec_ID)
 Unit OF Comm_Port (Port_ID)
 Unit OF Time_Zone (Time_Zone_ID)
 Unit OF RM_LCD_Messages (RM_LCD_Messages_ID)

RM_LCD_Messages

Index	Fields	Type
RM_LCD_Messages_ID	RM_LCD_Messages_ID	primary unique
RM_LCD_Messages_Name	RM_LCD_Messages_Name	unique

Order	Field Name	Data Type	Flags	Format
10	RM_LCD_Messages_ID	inte	im	->>>>>>>>>9
20	RM_LCD_Messages_Name	char	im	X(50)
30	Description	char		X(3000)
40	Date_Format	inte		>>9
50	Time_Format	inte		>>9
70	MSG1	char	m	X(16)
80	MSG2	char	m	X(16)
90	MSG3	char	m	X(16)
100	MSG4	char	m	X(16)
110	MSG5	char	m	X(16)
120	MSG6	char	m	X(16)
130	MSG7	char	m	X(16)
140	MSG8	char	m	X(16)
150	MSG9	char	m	X(16)
160	MSG10	char	m	X(16)
170	MSG11	char	m	X(16)
180	MSG12	char	m	X(16)
190	MSG13	char	m	X(16)
195	MSG14	char	m	X(16)
200	MSG15	char	m	X(16)
210	MSG16	char	m	X(16)
220	MSG17	char	m	X(16)
230	MSG18	char	m	X(16)
240	MSG19	char	m	X(16)
250	MSG20	char	m	X(16)
260	MSG21	char	m	X(16)
270	MSG22	char	m	X(16)
280	MSG23	char	m	X(16)
290	MSG24	char	m	X(16)

300	MSG25	char	m	X(16)
310	MSG26	char	m	X(16)
320	MSG27	char	m	X(16)
330	MSG28	char	m	X(16)
340	MSG29	char	m	X(16)
350	MSG50	char	m	X(16)
360	MSG51	char	m	X(16)
370	MSG52	char	m	X(16)
380	MSG53	char	m	X(16)
390	MSG54	char	m	X(16)
400	MSG55	char	m	X(16)
410	MSG56	char	m	X(16)
420	MSG57	char	m	X(16)
430	MSG58	char		X(16)
440	MSG59	char		X(16)
450	MSG60	char		X(17)
460	MSG61	char		X(17)
470	MSG62	char		X(16)
480	MSG63	char		X(16)
490	MSG64	char		X(16)
500	MSG65	char		X(16)
510	MSG66	char		X(16)
520	MSG67	char		X(16)
530	MSG68	char		X(16)

CCURE_Trac_Setup

Index	Fields	Type
Setup_ID	Setup_ID	primary unique
Setup_Name	Setup_Name	unique

Order	Field Name	Data Type	Flags	Format
10	Setup_ID	inte	i	->>>>>>>>>9
30	Setup_Name	char	im	X(50)
40	Description	char		X(3000)
50	Preset_Number	inte		->>9
60	Number_of_Slots	inte	m	9
70	Slot_1_Opening	inte		9
80	Slot_2_Opening	inte		9
90	Slot_3_Opening	inte		9
100	Slot_4_Opening	inte		9
110	Slot_1_Antenna	inte		9
120	Slot_2_Antenna	inte		9
130	Slot_3_Antenna	inte		9
140	Slot_4_Antenna	inte		9
150	Antenna_1_LED	inte		-9
160	Antenna_2_LED	inte		-9
170	Antenna_3_LED	inte		-9
180	Antenna_4_LED	inte		-9
190	Antenna_1_Direction	inte		9
200	Antenna_2_Direction	inte		9
210	Antenna_3_Direction	inte		9
220	Antenna_4_Direction	inte		9
230	Antenna_1_Camera_Output	inte		->9
240	Antenna_2_Camera_Output	inte		->9
250	Antenna_3_Camera_Output	inte		->9
260	Antenna_4_Camera_Output	inte		->9
270	Antenna_1_Camera_On_Any_Tag	logi		yes/no

280	Antenna_2_Camera_On_Any_Tag	logi	yes/no
290	Antenna_3_Camera_On_Any_Tag	logi	yes/no
300	Antenna_4_Camera_On_Any_Tag	logi	yes/no
310	Antenna_1_Camera_On_Forced_Door	logi	yes/no
320	Antenna_2_Camera_On_Forced_Door	logi	yes/no
330	Antenna_3_Camera_On_Forced_Door	logi	yes/no
340	Antenna_4_Camera_On_Forced_Door	logi	yes/no
350	Antenna_1_Camera_On_Tamper_DSM	logi	yes/no
360	Antenna_2_Camera_On_Tamper_DSM	logi	yes/no
370	Antenna_3_Camera_On_Tamper_DSM	logi	yes/no
380	Antenna_4_Camera_On_Tamper_DSM	logi	yes/no
390	Opening_1_DSM	inte	->9
400	Opening_2_DSM	inte	->9
410	Opening_3_DSM	inte	->9
420	Opening_4_DSM	inte	->9
421	Opening_1_DSM_Inverted	logi	yes/no
422	Opening_2_DSM_Inverted	logi	yes/no
423	Opening_3_DSM_Inverted	logi	yes/no
424	Opening_4_DSM_Inverted	logi	yes/no
425	Opening_1_DSM_Supervised	logi	yes/no
426	Opening_2_DSM_Supervised	logi	yes/no
427	Opening_3_DSM_Supervised	logi	yes/no
428	Opening_4_DSM_Supervised	logi	yes/no
430	Opening_1_REX	inte	->9
440	Opening_2_REX	inte	->9
450	Opening_3_REX	inte	->9
460	Opening_4_REX	inte	->9
461	Opening_1_REX_Inverted	logi	yes/no
462	Opening_2_REX_Inverted	logi	yes/no
463	Opening_3_REX_Inverted	logi	yes/no
464	Opening_4_REX_Inverted	logi	yes/no
465	Opening_1_REX_Supervised	logi	yes/no
466	Opening_2_REX_Supervised	logi	yes/no
467	Opening_3_REX_Supervised	logi	yes/no
468	Opening_4_REX_Supervised	logi	yes/no
470	Opening_1_PIR	inte	->9
480	Opening_2_PIR	inte	->9
490	Opening_3_PIR	inte	->9
500	Opening_4_PIR	inte	->9
510	Opening_1_PIR_Inverted	logi	yes/no
520	Opening_2_PIR_Inverted	logi	yes/no
530	Opening_3_PIR_Inverted	logi	yes/no
540	Opening_4_PIR_Inverted	logi	yes/no
550	Opening_1_PIR_Supervised	logi	yes/no
560	Opening_2_PIR_Supervised	logi	yes/no
570	Opening_3_PIR_Supervised	logi	yes/no
580	Opening_4_PIR_Supervised	logi	yes/no
590	Opening_1_DSO	inte	->9
600	Opening_2_DSO	inte	->9
610	Opening_3_DSO	inte	->9
620	Opening_4_DSO	inte	->9
630	Opening_1_EXO	inte	->9
640	Opening_2_EXO	inte	->9
650	Opening_3_EXO	inte	->9
660	Opening_4_EXO	inte	->9
670	Opening_1_Tag_Formats	inte	9
680	Opening_2_Tag_Formats	inte	9
690	Opening_3_Tag_Formats	inte	9
700	Opening_4_Tag_Formats	inte	9
710	Antenna_1_Aux_Input	inte	->9
720	Antenna_2_Aux_Input	inte	->9
730	Antenna_3_Aux_Input	inte	->9
740	Antenna_4_Aux_Input	inte	->9
750	Antenna_1_Aux_Input_Inverted	logi	yes/no
760	Antenna_2_Aux_Input_Inverted	logi	yes/no
770	Antenna_3_Aux_Input_Inverted	logi	yes/no
780	Antenna_4_Aux_Input_Inverted	logi	yes/no

790	Antenna_1_Aux_Input_Supervised	logi	yes/no
800	Antenna_2_Aux_Input_Supervised	logi	yes/no
810	Antenna_3_Aux_Input_Supervised	logi	yes/no
820	Antenna_4_Aux_Input_Supervised	logi	yes/no
830	Opening_1_Shunt	inte	->9
840	Opening_2_Shunt	inte	->9
850	Opening_3_Shunt	inte	->9
860	Opening_4_Shunt	inte	->9
870	Opening_1_Aux_Output	inte	->9
880	Opening_2_Aux_Output	inte	->9
890	Opening_3_Aux_Output	inte	->9
900	Opening_4_Aux_Output	inte	->9
910	Reader_Address	inte	>>9
920	Reader_Sync	inte	9
930	RS232_Comm	logi	yes/no
940	RS485_Comm	logi	yes/no
950	Gate_Delay	inte	>9
960	RS232_Comm_Delay	inte	>9
970	Tag_Charge_Time	inte	>9
980	Tag_Read_Time	inte	>9
990	Number_Doors	inte	9
1000	Number_Readers	inte	9
1010	Filespec	char	X(320)
1020	Text_Modified	logi	yes/no

Internal References

Table	Method	Phrase(s)
Time_Spec	Join – Time_Spec_ID	<ul style="list-style-type: none"> time_spec of unit
Time_Zone	Join – Time_Zone_ID	<ul style="list-style-type: none"> time_zone of unit
Comm_Port	Join – Port_ID	<ul style="list-style-type: none"> comm_port of unit
RM_LCD_Messages	Join – RM_LCD_Messages_ID	<ul style="list-style-type: none"> rm_lcd_messages of unit

Foreign References

Table	Method	Phrase(s)
Boards	Join - Unit_ID	<ul style="list-style-type: none">boards of unit
Comm_Port	Join - Port_ID	<ul style="list-style-type: none">unit of comm_port
Door	Join - Unit_ID	<ul style="list-style-type: none">door of unit
Inputs	Join - Unit_ID	<ul style="list-style-type: none">inputs of unit
Outputs	Join - Unit_ID	<ul style="list-style-type: none">outputs of unit
Reader	Join - Unit_ID	<ul style="list-style-type: none">reader of unit
Unit_Status	Join - Unit_ID	<ul style="list-style-type: none">unit_status of unit
Area	Join - Unit_ID	<ul style="list-style-type: none">area of unit
Elevator	Join - Unit_ID	<ul style="list-style-type: none">elevator of unit

Journal Object

Business Definition

A perpetual record of activity as generated by CCURE800.

Explanation

Only one instance of CF database exists; however, there can be multiple instances of the JOURN database. The objects stored in each of these JOURN databases reference objects in the CF database.

Journ Database Tables

Journ

Purpose:

- To record activity and audit trail as generated by CCURE800.

Index	Fields	Type
Local_DT	Local_DT	primary
Int_Data1	Int_Data1	active
Msg_Code	Msg_Code	active
User_PID	User_PID	active

Order	Field Name	Data Type	Flags	Format
30	Msg_Code	inte	i	>>>9
40	User_PID	inte	i	>>>, >>>, >>9
60	Txt_Data1	char		X(1000)
70	Txt_Data2	char		X(1000)
90	Local_DT	inte	i	>, >>>, >>>, >>9
100	Int_Data1	inte	i	>, >>>, >>>, >>9
110	Int_Data2	inte		>, >>>, >>>, >>9
120	Int_Data3	inte		>, >>>, >>>, >>9
130	Int_Data4	inte		>, >>>, >>>, >>9
140	Host_DT	inte		>, >>>, >>>, >>9
150	Jnl_ID	inte		>, >>>, >>>, >>9
160	TZ_Offset	inte		->, >>>, >>9

Manual_Action

Purpose:

- To record manual actions entered through a guard station. This table provides a permanent record of all actions entered for this journal volume rather than the transitory record(s) recorded in the GS_Action table of the CF database.

Index	Fields	Type
Deleted Action_ID	Deleted Action_ID	primary active

Order	Field Name	Data Type	Flags	Format
10	Action_ID	inte	i	->, >>>, >>9
20	Guard_ID	inte		->, >>>, >>9
30	Object_ID	inte		->, >>9.99
40	Action_Code	inte		->, >>>, >>9
50	Start_Time	inte		->, >>>, >>9
60	End_Time	inte		->, >>>, >>9
70	Priority	inte		->, >>>, >>9
80	Note	char		X(1000)
90	Deleted	logi	i	yes/no

Next_ID

Purpose:

- Used in place of a database sequence to assign the next journal entry ID number.

Index	Fields	Type
[default]	[none]	primary

Order	Field Name	Data Type	Flags	Format
10	Next_Journal_ID	inte		->, >>>, >>9

JNL_Object

Purpose:

- Not used at this time. Reserved for future use.

Index	Fields	Type
	Object_ID	primary

Order	Field Name	Data Type	Flags	Format
10	Object_ID	inte	i	>>>>>9
20	Name	char		X(20)
60	Type_IDX	inte		>>9

JNL_Links

Purpose:

- Not used at this time. Reserved for future use.

Index	Fields	Type
	Source_ID	primary
	Target_ID	active

Order	Field Name	Data Type	Flags	Format
10	Source_ID	inte	i	>>>, >>9
20	Target_ID	inte	i	>>>, >>9
30	State_IDX	inte		>, >>9
40	Action_IDX	inte		>, >>9
50	Enabled	logi		yes/no
60	Start_DT	deci-2		->>, >>9.99
70	End_DT	deci-2		->>, >>9.99

Foreign References

- No direct foreign references *
- Journ.User_PID and Manual_Action.Guard_ID refer to Person.Person_ID in the CF database.
- Other fields in Journ table contain values that depend on the message type of the entry.
- The most common foreign reference is to the Obj_Archive table in the CF database where the value(s) in the Journ table entry may reference an existing or deleted object.

Reference Tables

Business Definition

A set of tables not maintainable by or visible to the user, which supports various operations performed by the application software.

Convention

- No specific conventions or usage rules are established at this time.

Primary Tables

Jnl_Directory

Purpose:

- To maintain a perpetual record of all journal databases created by CCURE800.
- The Volume field contains the number of the current Journal Database file.

<u>Index</u>	<u>Fields</u>	<u>Type</u>			
Volume	Volume	primary unique			
Current_Volume	Current_Volume	active			

Order	Field Name	Data Type	Flags	Format
10	Volume	inte	i	->>>>>>>>9
20	Comments	char		x(3000)
30	Directory	char		x(320)
40	Backup_Media	char		x(1)
50	Current_Volume	logi	i	yes/no
60	Exists_on_disk	logi		yes/no
70	First_Event_DT	inte		->, >>>, >>>, >>9
80	Last_Event_DT	inte		->, >>>, >>>, >>9
90	Journal_Created_DT	inte		->, >>>, >>>, >>9
100	Journal_Closed_DT	inte		->, >>>, >>>, >>9
110	Backup_Date	date		99/99/9999
120	Backup_Time	inte		>>>>9
130	Backup_Directory	char		X(320)
140	DB_Version	inte		->, >>>, >>9
150	Service	char		X(16)
160	Hostname	char		X(32)
170	Backup_Complete	logi		yes/no

Jnl_Replay_Select

Purpose:

- Table used to pass journal replay selection data from Admin client to Coordinator Progress journal replay client.

Index	Fields	Type
Jnl_Replay_ID_Item_ID	Jnl_Replay_ID Item_ID	primary unique

Order	Field Name	Data Type	Flags	Format
10	Jnl_Replay_ID	inte	im	>>>>>>>>9
20	Item_ID	inte	im	>>>>>>>>9
30	Sel_Criteria	inte		>>9

License_Item

Purpose:

- Contains licensing information as it pertains to the CCURE800 installation at this customer site.

Index	Fields	Type
License_Item_Name	License_Item_Name	primary unique

Order	Field Name	Data Type	Flags	Format
10	License_Item_Name	char	im	X(20)
20	Value_Int1	inte		->>>>>>>9
30	Value_Int2	inte		->>>>>>>9
40	Value_Int3	inte		->>>>>>>9
50	Value_Char1	char		X(320)
60	Value_Log1	logi		yes/no
70	Value_Dat1	date		99/99/9999
80	Expiration_date	date		99/99/9999
90	Activation_date	date		99/99/9999
100	Authorization_Checksum	inte		->>>>>>>9
110	Disabled_by_user	logi		yes/no

Obj_Archive

Purpose:

- To maintain a perpetual record of the names for all objects referenced in all journal (journ) tables.

Index	Fields	Type
Object_ID	Object_ID	primary unique
Object_Name	Name	active
Object_Type	Object_Type	active

Order	Field Name	Data Type	Flags	Format
10	Object_ID	inte	i	->>>>>>>>>9
20	Name	char	i	X(50)
30	Deleted	logi		yes/no
40	Object_Type	inte	i	>>9

Internal References

Table	Method	Phrase(s)
Map_Icon	Join - Object_ID	<ul style="list-style-type: none"> map_icon of obj_archive
GS_Action	Join - Object_ID	<ul style="list-style-type: none"> gs_action of obj_archive
Group_Member	Join - Object_ID	<ul style="list-style-type: none"> group_member of obj_archive
All object tables	Join - Object_ID	
Journ	Join - Object_ID	<ul style="list-style-type: none"> Join to any field using an object ID.

Obj_Name

Purpose:

- To assure all objects have a unique name system wide.

Index	Fields	Type
Name	Name	primary unique

Order	Field Name	Data Type	Flags	Format
10	Name	char	im	X(20)
20	Object_ID	inte		>>>>>>9
30	Object_Type	inte		>>9

Table	Method	Phrase(s)
Obj_Archive	Join - Name	<ul style="list-style-type: none"> obj_archive of obj_name

Messages

Purpose:

- Contains error message text displayed to the user in each supported language.

Index	Fields	Type
Msg_ID_Lang	Msg_ID Language	primary unique
Msg_ID	Msg_ID	active
Msg_Name	Msg_Name	active

Order	Field Name	Data Type	Flags	Format
10	Msg_Name	char	im	x(32)
20	Msg_ID	inte	im	->>>>>>>>9
30	Msg_Text	char	m	x(3000)
40	Description	char		x(3000)
50	Language	inte	im	>>>9
60	Last_Mod_DT	inte		-> ,>>> ,>>> ,>>9

Sequence

Purpose:

- Reserved for the assignment sequence numbers and the storage of incidental control values.

Index	Fields	Type
Type	Type	primary unique
Order	Field Name	Data Type
10	Type	char
20	Last_Value	inte

Table	Method	Phrase(s)
Reader	Join - Type	• reader of sequence

Sysops

Purpose:

- Reserved for storing various system settings that may apply to this particular installation.

Index	Fields	Type
Var_Num	Var_Num	primary unique
Order	Field Name	Data Type
10	Var_Num	inte
50	VarInt	inte
60	VarChar	char
70	VarLog	logi

CHUID_Format

Purpose:

- Reserved for storing the CHUID (CardHolder Unique ID) Format that applies to this particular installation.

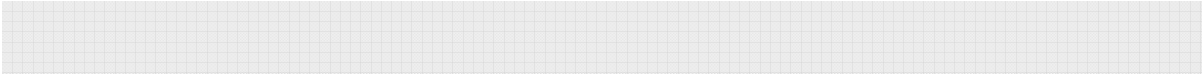
Table: CHUID_Format

Order	Field Name	Data Type	Flags	Format
10	CHUID_Format_ID	inte	im	->, >>>, >>>9
20	CHUID_Format_Name	char	m	x(50)
30	Description	char		X(3000)
40	Update_ID	inte		->>>>>>>>>9
50	CHUID_Length	inte		>>>>>>>>9

Table: CHUID_Format_Field

Order	Field Name	Data Type	Flags	Format
10	Field_ID	inte	im	->>>>>>>>>9
20	CHUID_Format_ID	inte	im	->, >>>, >>>9
30	Field_Position	inte		>>>>>9
40	Field_Length	inte		>>>>>9
50	Field_Order	inte		>>>>>9
60	Field_Label	char		X(50)

Chapter 4



Accessing Journal Information

This chapter describes the structure of the journal tables used in the C•CURE System database, and explains how to look up journal information.

In This Chapter

- How the Journal Database is Organized
- Looking up Journal Information
- Compressed Integer Date/Time Format
- Decoding the Compressed Integer Date/Time Value
- Journal Table Structure and Message Codes

How the Journal Database is Organized

Real-time information about card accesses and other recorded system activity is stored in the JOURN table of the Journal database, in a file called JL_XXXXX.db (where XXXXX is the volume number of the journal.)

- The JNL_Directory table in the CF database stores information about which journal file to use for looking up information about card accesses on a particular date. This JNL_Directory table contains one record for each journal volume created. The current_volume field is set to TRUE if the record refers to the journal volume (JL_XXXXX) currently being used.
- The *first_event_dt* and *last_event_dt* fields contain the first and last *local* times of events recorded in the particular journal file. Local times refer to times when the event occurred at the panel.
- The last event date of the active Journal is not updated until it is switched.
- All times are stored as elapsed seconds since January 1, 1990, GMT

For example, clearance information for each cardholder is stored in the CF database. However, real-time information about card accesses is stored in the JOURN table of the JOURN database, which is stored in a file called JL_XXXXX.db, where XXXXX is the volume number of the current journal.

The C•CURE 800/8000 Driver coordinator automatically switches journal files when the current one is full and numbers them sequentially, beginning with JL_00001.

When the journal database server is started, it is connected to a TCP/IP port. One of two ports is used, depending on whether the journal files are odd-numbered or even-numbered.

- Odd numbered journal files are connect to port JNSRV1
- Even numbered journals are connected to port JNSRV2

Looking up Journal Information

Each record in the Journal table reports one journal message. The date stored in the record varies depending on the type of Journal message reported. The Message tables in this chapter list the message types found in the table and the contents of each data field for an individual message type.

Journal information in the JOURN database is tracked by message code that is stored in the `Msg_Code` field, and Person Id, which is stored in the `User_Pid` field.

For example, when `Msg_Code = 2` (card admitted) for a particular user id, then `INT_Data1` contains the reader id of the reader used and `INT_Data2` contains the admit code. The reader id can then be used to find information about the corresponding door in the CF database.

Finding the Current Journal Database

The `Volume` field in the `CF.JNL_DIRECTORY` table contains the number of the current Journal Database file (`JL_xxxx.db`)

For example, if `JNL_DIRECTORY.VOLUME=2`, then the current journal database is named `JL_00002.DB`, as shown in the following diagram.

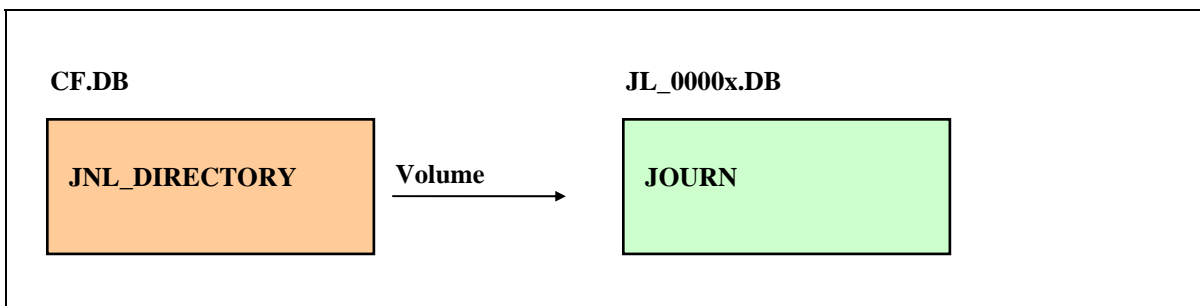


Figure 29: Volume Field in `JNL_DIRECTORY` table points to `JL` database

Compressed Integer Date/Time Format

Progress supports a standard Date data type. All times are stored as a standard integer value, that is, the number of seconds elapsed since midnight.

C•CURE 800/8000 uses a special compressed integer date/time format for time-stamping journal messages, which uses less space and is sortable.

The compressed date/time integer is created as follows:

1. Convert the date when the activity occurred to the number of days since January 1, 1990.
2. Multiply this value by 86400, the number of seconds in a day.
3. Add in the time (number of seconds since midnight when the activity occurred.)
4. Convert this value to Greenwich Mean Time by subtracting the number of seconds that the local time (where the journaled activity occurred) is offset from GMT.
5. The compressed date/time integer (number of seconds elapsed past January 1, 1990 when the event occurred) is stored in the **Host_Date** field in the **Journ** table.

Decoding the Compressed Integer Date/Time Value

This section contains two sample procedures that you can use to convert the compressed integer date/time value in the **Host_Date** field from seconds to a standard date/time value.

- **(A) Sample Progress Procedure to Decode Compressed Date/Time Value** – Sample procedure written in Progress to decode any journal date/time value.
- **(B) Sample using Microsoft Access/Excel to Decode Compressed Date/Time Value** – Sample procedure using a Formula and Microsoft Access and/or Microsoft Excel to decode the compressed date/time value.

A. Sample Progress Procedure to Decode Journal Date/Time Value

Following is a sample procedure written in Progress that can decode the journal date/time and time zone integers into a Local Time integer and date for the time zone in which the event occurred.

PROCEDURE PRC\$DecodeJnlLocalDT:

```

/* -----
Purpose:
    Decode the Journal date/time and time zone integers into a Local
    Time integer and date for the time zone in which the event occurred.

Parameters:
input:  INT$Date_Time   - Date/Time integer from journal file (GMT).
        INT$Time_Zone  - Offset of local time from GMT in 15 minute
                        increments (e.g. 1 = 15 minutes). We assume
                        (+) values are after GMT (+48 is Wellington)
                        and (-) are before GMT (-20 is Eastern
                        Standard Time).

output: DAT$Date       - Date of activity in local time.
        INT$Time       - Time of activity in local time.

----- */

/* Input/Output Parameters. */

DEFINE INPUT PARAMETER INT$In_Date_Time   AS INTEGER.
DEFINE INPUT PARAMETER INT$Time_Zone     AS INTEGER.

DEFINE OUTPUT PARAMETER DAT$Date         AS DATE.
DEFINE OUTPUT PARAMETER INT$Time         AS INTEGER.

/* Local variables. */

DEFINE VARIABLE INT$Days                 AS INTEGER.
DEFINE VARIABLE INT$Date_Time            AS INTEGER.

DEFINE VARIABLE INT$One_Day              AS INTEGER.
DEFINE VARIABLE INT$Increment_Value      AS INTEGER.
DEFINE VARIABLE DAT$Begin_Date           AS DATE.

/* Initialize variables. */

ASSIGN
    INT$One_Day      = 86400                /* Number of seconds in a day. */
    DAT$Begin_Date  = 1/1/90              /* Base date used for journal dates. */
    INT$Increment_Value = 900.            /* Number of seconds in 15 minutes. */

/* Convert the integer date/time to local time from GMT. */

ASSIGN INT$Date_Time = INT$IN_Date_Time + (INT$Time_Zone * {&INCREMENT_VALUE}).

/* Get the time as the number of seconds since midnight. */

ASSIGN INT$Time = (INT$Date_Time ) MODULO {&ONE_DAY}.

/* Get the date as the number of days since our base date (1/1/90). */

ASSIGN INT$Days = (INT$Date_Time - INT$Time ) / {&ONE_DAY}.

/* Convert this offset to a regular date. */

ASSIGN DAT$Date = INT$Days + {&BEGIN_DATE}.

END PROCEDURE.

```

B. Sample Procedure using Microsoft Access/Excel to Convert Date/Time Value

To convert the **Host_Date** value in the **journal** table in the Journal database from seconds and have it present itself in a standard Date/Time format, the following formula will work in Microsoft Access.

$$= (([\text{Host_Date}]/864000)+(0.0416666666666667)+[\text{Test1}]) - 0.208333333333333$$

Components of the formula are:

Field	Description
Host_date	Value in the Host_Date field in the Journ table within the Journal database. This number represents how many seconds past January 1, 1990, the event occurred.
86400	Number of seconds in a day, which is 86400 .
0.0416666666666667	An adjustment number based on whether or not you are observing Daylight Savings Time. If observed, Daylight Savings Time is a 1-hour difference. Therefore, this number is derived as follows: Number of seconds in an hour = 3600 Number of seconds in a day = 86400 Percentage of a day to adjust when observing DST = 0.0416666666666667 (3600/86400 = 0.0416666666666667)
Text1	Label of a field in Microsoft Access containing the value "January 1, 1990", also known as the "base date" since all dates represent the number of seconds AFTER this base date that an event occurred in the Journal database. Microsoft Access and Microsoft Excel convert this "base date" to a number. You can also plug in the numerical equivalent of January 1, 1990 into the formula, which is 32874
0.208333333333333	This number represents an adjustment based on the OFFSET to GREENWICH MEAN TIME (GMT) of the Time Zone where you are or where the CCure800/8000 host is located. Example: Eastern Standard Time (EST) has a GMT offset of 5 hours. There are 18000 seconds in 5 hours, so the percentage of a day that you adjust to compensate for GMT Offset = 0.208333333333333 (18000/86400=0.208333333333333)

Example of Using the Formula to Convert from Seconds to Date/Time Value

If the **Host_Date** field in the Journ table contains the date in seconds, for example, **365028282**, you can use the formula to calculate the actual date/time value as follows:

1. $((365028282/864000)+(0.0416666666666667)+[32874]) - 0.208333333333333$
2. $(4224.864375+.0416666666666667+32874) - 0.208333333333333$
3. $37098.90604 - 0.208333333333333$
4. 37098.69771

5. Plug the final result into Microsoft Excel or Microsoft Access and change the format of the field or cell into a Date/Time format. For example, if you enter *37098.69771* into a cell and select the Date/Time format, the result is July 26, 2001, 4:44 PM

Journal Table Structure and Message Codes

The following table indicates how all journal fields are interpreted.

Journal Table Fields

Field-Name	Data Type	Index	Format	Initial	Description
Jnl_ID	integer	Yes	>, >>>, >>>, >>>9	Seq.	Unique ID for message (max of 2 billion)
Local_DT	integer	Yes(P)	>, >>>, >>>, >>>9	0	Encoded Date/Time when activity actually occurred
Host_DT	integer	No	>, >>>, >>>, >>>9	0	Encoded Date/Time that message was received at host
TZ_Offset	integer	No	->, >>>, >>>9	10	Time zone offset in half-hours
Msg_Code	integer	Yes	>>>>9	0	Message code for activity
User_PID	integer	Yes	>>, >>>, >>>9	0	PID of person associated with activity
Int_Data1	integer	Yes	>, >>>, >>>, >>>9	0	May only contain object IDs
Int_Data2	integer	No	>, >>>, >>>, >>>9	0	May contain either object IDs, or integer < 1000
Int_Data3	integer	No	>, >>>, >>>, >>>9	0	May contain either personnel Ids (PIDs), or integer < 1000
Int_Data4	integer	No	>, >>>, >>>, >>>9	0	May not contain object IDs or PIDs that need to be searched as selection criteria for journal replay.
Txt_Data1	character	No	X(80) (varying)		Message-specific text string
Txt_Data2	character	No	X(80) (varying)		Another message-specific text string

NOTE:

- To optimize querying of Int_Data fields, Personnel IDs and Object IDs must start at 1000 or higher. That way, it is easy to distinguish between an Int_data field containing a small integer (<1000) and one containing an object ID (>1000)

- Int_Data1 and Int_Data2 are restricted to enable fast searches on these fields for object IDs. These are the only fields that should ever contain Object Ids which can be searched on.

- In Net Video activity, Int_Data3 stores integers for Net video use only, for example, Int_Data3 = 7 indicate it is a NetVideo tour pattern. (see jnl_parm.i)

- Int_data4 usually contains general numeric data; this data is NEVER used to select journal entries to replay. Therefore, if it contains an Object_ID, the object must NOT be considered necessary to specify the journal object which you select on, since you will not be able to do that selection. An example of a case where it would be OK to use Object_ID in data4 would be to include the object_ID of the causer object, since that is not part of the object being replayed, and will not be selected. See message code 20. Only use an Object_ID in int_data4 if absolutely necessary.

- Date/Times are converted to GMT and stored as an encoded value (in seconds from Jan. 1, 1990) .

Message Codes and Descriptions

Parameters listed below are from *jnl_parm.i* and *jns_parm.i*, unless indicated otherwise. An asterisk preceding a number indicates a note. For example, *1 refers to Note *1 in the following **Message Code Notes** table.

Msg Code	Description	User_PID Supplied	Int Data	Txt Data
001	User login/logout	PID User	(1) none (2) Program started - PRM\$JPR_xxx (3) Login/out code - PRM\$JLO_xxx (4) none	(1) Node (2) Username *1
002	Card admitted	PID	(1) Door ID (2) Admit code *4 (3) SO `*2 (UNUSED) (4) cardNumber	(1) Escort code (PM\$JEA) Extended Reader Msg *39,*4 (2) Escort ID “ ”PersonName, if temp flag used “ ”CHUID if extended card *55 ‘ ’ is separator *40
003	Card rejected	PID	(1) Door ID (2) Admit code *4 (3) Reject code - PRM\$JRE_xxx (4) cardNumber	(1) Escort code (PRM\$JEA) “ ”Extended Reader Msg *39,*44 (2) PersonName, if temp flag used “ ”CHUID if extended card *55
004	Log message	SO *2 PID User	(1) Event object ID *3 (2) none (3) none (4) unused (was JNL ID of related activity *3)	(1) Text of log message (2) none
005	Object changed state - Event/distributed/manual *7 *8	PID *21	(1) ID of object changing state (2) ID of object Id *53 (3) State code *5 (4) State change method code *20 or code for iStar connection	(1) none *51 (2) none

Msg Code	Description	User_PID Supplied	Int Data	Txt Data
006	Manual action *8 *9	SO PID User	(1) ID of object acted on (PID of Graced card *41) (2) action code *6 (3) Manual action object ID *10 (4) manual action - PRM\$JMA_XXX	(1) count and personnelgroupID *42 (2) PID of area lockout grace *47
007	System activity (normal)	No	(1) jnl_export_id *46 (2) none (3) Activity code - PRM\$JSM_XXX (4) none	(1) Node name (2) MAC name *23
008	System error	No	(1) jnl_export_id *46 (2) none (3) Sys. error code - PRM\$JSE_XXX (4) API error code *11	(1) Node name (2) API name or missing jnls for journal export msgs *3
009	Device activity (normal)	PID *22	(1) ID of unit or component *24, *32, *34, *49 (2) anotherObjectId *14, *25, *33, *35, *50 (3) Activity code - PRM\$JDM_XXX (4) none	(1) firmware version *16, *26, *36 (2) none
010	Device error/recovery	No	(1) ID of unit or component (2) anotherObjectId *14 (3) Error code - PRM\$JDE_XXX (4) subErrorCode (paging) - PRM\$WOERR	(1) firmware version *16 (2) none
011	AssetActivity	AssetId *12	(1) none (2) Asset act. code - PRM\$JAT (3) PersonId (4) Asset access code - PRM\$JAA *15	(1) none (2) none
012	AssetMovement Authorized	AssetId *12	(1) DoorId (2) HHRId (3) PersonId (4) Asset access code - PRM\$JAA, PRM\$JRC *13	(1) tag number (2) Area Id *17
013	AssetMovement Unauthorized	AssetId *12	(1) DoorId (2) HHRId (3) PersonId (4) Asset access code - PRM\$JAA, PRM\$JRC *13	(1) tag number (2) Area Id *17

Msg Code	Description	User_PID Supplied	Int Data	Txt Data
014	AssetMovement Attempted	AssetId *12	(1) DoorId (2) (none) (3) PersonId (4) Asset access code - PRM\$JAA, PRM\$JRC *13	(1) tag number (2) Area Id *17
015	Asset Location Update	AssetId *12	(1) AreaId (2) HHR Id *18 (3) PersonId *18 (4) Asset access code - PRM\$JAA *19	(1) tag number (2) none
016	Watchtour Action	PID	(1) WT action code - PRM\$ACT *6 (2) ObjectId (3) TourGuardId	(1) none (2) none
017	Watchtour Activity	NO	(1) WT act. code - PRM\$JWTACT (2) ObjectId *31 (3) TourGuardId (4) ReaderId	(1) none *30 (2) none
018	Watchtour Error	NO	(1) WT end code - PRM\$JWTE *30 (2) ObjectId *31 (3) TourGuardId	(1) TourStop name list or none *54 (2) none
019	Watchtour Stop Activity	NO	(1) WT stop code - PRM\$WJTS *30 (2) ObjectId *31 (3) TourguardId (4) TourStopId	(1) none (2) none
020	NetVideo Activity	PID *29	(1) CameraId (2) NetVideoActionId *45 (3) NetVideo Code – PRM\$JNVA (4) EventId *28	(1) SegmentId *27 (2) none
021	Keypad Command Activity	PID	(1) Keypad activity code - PRM\$JRE (2) Door Id (3) Keypad Cmd Object ID * 48 (4) Direction - 0=IN, 1=OUT	(1) Card number (2) Event ID
022	Intrusion Zone Activity	NO	(1) IZone ID (2) IZ act. code - PRM\$JIZACT (3) none (4) none	(1) one or more object Ids or none *37 (2) Person Id *38
023	Intrusion Zone Error	NO	(1) IZone ID (2) IZ error code - PRM\$JIZER (3) none (4) none	(1) one or more object Ids or none *37 (2) Person Id *38

Msg Code	Description	User_PID Supplied	Int Data	Txt Data
024	Area Activity	PID *52	(1) Area ID (2) Personnel group ID (3) Area act. code – PRM\$JAREA (4) Count *43	(1) none (2) none

**** Message Code Notes ****

This section explains the Notes in the **Message Codes and Descriptions** list. In the list, Notes are indicated by a numbered asterisk, for example, *1 refers to Note *1 (Username is always provided, even for valid logins) as shown below.

- *1 Username is always provided, even for valid logins.
- *2 SO - Personnel ID of Security Officer. Supplied only if a manual admit was performed.
- *3 Argument is optional. It is only supplied in two cases; First, if there is an API error. Second, for enhanced reporting Missing Journal errors, the text will display a list of the missing journals separated by spaces) e.g.: [11 22 33 44].
- *4 Cardholder admit codes (PRM\$JAD_XXX) - The flags are additive. Note that only some codes are valid on card rejects.
For reader direction the bits used in the admit code are
 - PRM\$JAD_Door_Unused = 1, (non-zero based) /* Door was not used */
 - PRM\$JAD_Direction_IN = 7, (non-zero based) /* direction of read was IN */
 - PRM\$JAD_Direction_OUT = 8, (non-zero based) /* direction of read was OUT */
 - PRM\$JAD_Ext_Reader_Status_Msg = 9, (non-zero based) /* extended reader status */
- *5 New state code - The new state of the specified object. See PRM\$OST parameters in LNK_PARM.I. If action is a pulse, then the state simply indicates the type of pulse.
- *6 Action code indicates the action that occurred. See PRM\$ACT parameters in LNK_PARM.I.
- *7 Message code 5 originally had a deferral code (PRM\$JDF_XXX) to indicate that the action did not take place immediately. This deferral code has been removed.
- *8 The object state change message is generated at the time the action actually takes place (i.e. not necessarily at the time the action is scheduled). This can be affected by a start time associated with a manual action, and by competing actions of higher priority.
- *9 The manual action message is generated in four possible cases: when a manual action is scheduled, when it is activated, if it is canceled, and when it deactivates. If the scheduled and activated times of a manual action are the same, only the activated message is signaled. If the manual action is canceled, the deactivated message is not signaled. The manual action is the only message type that may contain the object ID of a group. When a manual action goes active, and when it is deactivated or canceled, object state change messages (type 5) are generated for every object changing state (e.g. one message for each object in a group).
- *10 Additional manual action information is stored in the manual action table. This includes the scheduled start and end date/time of the manual action, the ID of the object or object group changing state, and the state code (see note 5).
NB: This field should NOT have contained any value >1000, since it does not contain a personnel id. Special journal replay code had to be added to fix this bug.
- *11 API error codes are displayed and an integer value. The explanation of an error code may be found by looking up the API name in the Microsoft API reference guide.
- *12 AssetId is taken from the same pool of sequences as PersonId, so that it won't overlap with Personnel Ids. This is the main id for this type of message, so we are using the PID field.

- *13 AccessCode: Currently this field is split into two parts
Hi 3 Byte Admit Code PRM\$JAA_XXX This is a bitmap of admit flags
1st Byte RejectCode PRM\$JRC_XXX This is not a bitmap.
- *14 This field typically contains comm port id or modem id. It is set to 0 for many device status messages.
- *15 AccessCode: Admit Code PRM\$JAA_XXX This is a bitmap of admit flags
- *16 Firmware version: Use for messages about flash ROM download
- *17 For asset movement messages, the Area Id is sent in data2, to 4gl server and used to update the asset location. But it is not Journalled.
- *18 If Handheld reader did update, Data 1 will have the HHR Id. If an Admin user did the update, Data3 will have the PersonId.
- *19 Same as *13, but the RejectCode part will not be used. Admit flags used will be Stationary, Noticed.
- *20 A state change method code will stored in the int_data4 field. This value is defined as an enum ObjectStateMethodCode in 4gl_defs.h/cpp_defs.i. It is one of the PRM\$CSC_XXX values used to define what had caused the state to change. This field should be set to 0 if it is not used. This field is also used by the state change messages for iStar communication. see 4gl_defs.h PRM\$VIA_XXX values.
- *21 The User_PID field will be used to store the Person ID of the person who caused the state to change for state change messages. If the person is not known or applicable, this field should be set to 0.
- *22 PID of the card that was graced, for Grace Card Device Activity messages.
- *23 Argument is optional. The MAC address is only supplied if an iStar panel was rejected by the host when attempting to connect.
- *24 For Radionics, this field typically might contain Radionics Receiver id.
- *25 For Radionics, this field typically might contain Radionics Account id.
- *26 For Radionics, this field typically might contain information on area number and point number.
- *27 For V7.0, NetVue, uses txt_data(1) to store Video Segment string, for translation, see NVTools.cpp/h
- *28 For NetVideo, if netvideo action is triggered by an event, this field is used for an event Id.
Note * Since this field is stored in IntData_4, the journal replay cannot search on this event.
- *29 For NetVideo manual recording, this field is used to store the Person ID of the person who triggers netvideo recording.
- *30 For GuardTour activities, PRM\$JWTACT_XXX
- *31 Watchtour ID.
- *32 For NetVideo, this field contains action id, camera id, or server id.

- *33 For NetVideo, this field typically might contain action id, camera id, or server id.
- *34 For BiDirectional, this field typically might contain receiver id.
- *35 For BiDirectional, this field typically might contain action id or receiver id.
- *36 For BiDirectional, this field typically might contain unknown message or unknown Ack message from a device.
- *37 For Intrusion Zone Activity and Error messages the text1 field contains object ids for the chain of causer objects of the zone's state change. This list might contain event ids, keypad command ids, and input ids in a comma separated list.
- *38 For Intrusion Zone Activity and Error messages the text2 field may contain a person ID as the originating causer of the zone's state change. Person Ids are stored in this field instead of the normal PID field as this Person Id is part of the cause list stored in the Text 1 field.
- *39 Escort code could be visitor, escort or escorted visitor. Defined in jnl_parm.i, entry is PRM\$JEA_XXX.
- *40 Escort id is the id for escort if the person is escorting. Card Reject msg had this field but journal replay never referenced this field for CC800 Versions 7.1 and 7.2. For card reject msg, this field isn't used at all from CC800 Version 8.0 on. This field is also shared with person name if temp flag is turned on. It is separated from the Escord ID by a "|" pipe.
- *41 When manual action is a "Grace Card" this field will contain the PID of the graced person record. This means that query on object may bring up some personnel records, and that query on person won't bring up this record.
- *42 Count of people to be set in the area + divider '|' + personnel group id.
- *43 Count of people to be set in the area.
- *44 Escort Code(size:2) + divisor '|' (size:2) + extended reader status message (max:72).
- *45 Either Action Id for event triggered netvideo action or 0 for netvideo manual action.
- *46 For System Errors or System Activity related to enhanced reporting, the jnl_export_id will be stored in int_data1
- *47 Person ID of Area lockout grace card is stored in TEXT2. Note that Person ID of regular grace card is stored in INT1 which has a problem that query on person won't bring this PID described in *41.
- * 48 Int_data3 should contain only Person_Ids, not Object_Ids. This may create issues in journal replay selection.
- *49 For Demuster, this field will either contain area id or unit id depending on which demuster command is requested, demuster all or demuster area.
- *50 For Demuster, this field will contain either partition id or nothing.
- *51 For GTAA, this field will contain input names separated by "," character

- *52 This field will contain personnel ids that represent personnel who make pass-through violations in an area.
- *53 The field will be used to store some related-object id of object changing state, e.g. for inovonics, receiver object id when gateway input gets activated. If this field is not used, the field will be set to 0.
- *54 For guard tour error messages, the text1 field contains tourstop names for missed stops or none in a ‘|’ separated list.
- *55 For access messages with extended cards, we store the CHUID in text2, preceded by “|”.
For admits, it will be the 3rd piece of data and must be preceded by 2 “|”.
For rejects it will be the 2nd piece of data and must be preceded by “|”.

Journal Message Codes

Following is a list of parameters used to interpret the various journal codes.

```

/*
  Journal CCURE 8000 program name codes. Defines all journal CCURE 8000 program names.
  Used in the program name field of the PRM$JMC_Loginout message.
*/
/* enum JournalProgramNameCode */
*/
&GLOBAL-DEFINE PRM$JPR_apC_Driver 1 /* apC driver */
&GLOBAL-DEFINE PRM$JPR_Coordinator 2 /* Coordinator */
&GLOBAL-DEFINE PRM$JPR_Monitoring_Station 3 /* Monitor (event or activity) */
&GLOBAL-DEFINE PRM$JPR_Map_Station 4 /* Map station */
&GLOBAL-DEFINE PRM$JPR_Configuration 5 /* Configuration program */
&GLOBAL-DEFINE PRM$JPR_Activity_Printer 6 /* Activity printer application */
&GLOBAL-DEFINE PRM$JPR_Third_Party_GS 7 /* Third party GS */
&GLOBAL-DEFINE PRM$JPR_Last_Program 7 /* Should always equal the highest code */
*/

  Journal user login/logout codes. Defines all journal login/logout codes. Used in
  the login code field of the PRM$JMC_Loginout message.
*/
&GLOBAL-DEFINE PRM$JLO_User_Logged_In 1 /* User logged in successfully */
&GLOBAL-DEFINE PRM$JLO_Login_Attempt_Rejected 2 /* User login attempt failed */
&GLOBAL-DEFINE PRM$JLO_User_Logged_Out 3 /* User logged out */
&GLOBAL-DEFINE PRM$JLO_GSDisconnected 4 /* GS lost tcp connection */
&GLOBAL-DEFINE PRM$JLO_Last_Loginout 4 /* Should always equal the highest code */
*/
/*
  Journal cardholder admit codes. Defines all journal admit codes. Used in the
  admit code field of the PRM$JMC_Card_Admit and PRM$JMC_Card_Reject messages.
  These flags are additive, thus each flag is a factor of two.
*/
&GLOBAL-DEFINE PRM$JAD_Door_Unused 1 /* Door was not used */
&GLOBAL-DEFINE PRM$JAD_Noticed 2 /* Card is flagged as noticed */
&GLOBAL-DEFINE PRM$JAD_Duress 3 /* Cardholder used duress mode */
&GLOBAL-DEFINE PRM$JAD_Host 4 /* Admit processed by host */
&GLOBAL-DEFINE PRM$JAD_Manual 5 /* Manual admit by guard */
&GLOBAL-DEFINE PRM$JAD_Deleted 6 /* Card flagged as deleted */
&GLOBAL-DEFINE PRM$JAD_Direction_IN 7 /* direction of read was IN */
&GLOBAL-DEFINE PRM$JAD_Direction_OUT 8 /* direction of read was OUT */
&GLOBAL-DEFINE PRM$JAD_Ext_Reader_Status_Msg 9
&GLOBAL-DEFINE PRM$JAD_group_access 10 /* carpool group access */
&GLOBAL-DEFINE PRM$JAD_PINOnlyAccess 11 /* PIN Only access */
&GLOBAL-DEFINE PRM$JAD_ComplexCHUID 12 /* Extended Card access */
&GLOBAL-DEFINE PRM$JAD_Last_Card_Admit 12 /* Should always equal the highest code */
*/

/* Bit Masks. Following are bit masks for the above admit codes. */

&GLOBAL-DEFINE PRM$JAD_B_Door_Unused 1 /* Door was not used */
&GLOBAL-DEFINE PRM$JAD_B_Noticed 2 /* Card is flagged as noticed */
&GLOBAL-DEFINE PRM$JAD_B_Duress 4 /* Cardholder used duress mode */
&GLOBAL-DEFINE PRM$JAD_B_Host 8 /* Admit processed by host */
&GLOBAL-DEFINE PRM$JAD_B_Manual 16 /* Manual admit by guard */
&GLOBAL-DEFINE PRM$JAD_B_Deleted 32 /* Card flagged as deleted */
&GLOBAL-DEFINE PRM$JAD_B_Direction_IN 64 /* direction of read IN */
&GLOBAL-DEFINE PRM$JAD_B_Direction_OUT 128 /* direction of read OUT */
&GLOBAL-DEFINE PRM$JAD_B_Ext_Reader_Status_Msg 256 /* Extended reader status message */
&GLOBAL-DEFINE PRM$JAD_B_group_access 512 /* Carpool group access */
&GLOBAL-DEFINE PRM$JAD_B_PINOnlyAccess 1024 /* PIN Only access */
&GLOBAL-DEFINE PRM$JAD_B_ComplexCHUID 2048 /* Extended Card access */
*/

```

```

/*
  Journal cardholder reject codes. Defines all journal reject codes. Used in the
  reject code field of the PRM$JMC_Card_Reject message.

*/
&GLOBAL-DEFINE PRM$JRE_Admit 0 /* Card was admitted */
&GLOBAL-DEFINE PRM$JRE_Unknown_Card 1 /* Card not in database */
&GLOBAL-DEFINE PRM$JRE_Clearance 2 /* Clearance rejected */
&GLOBAL-DEFINE PRM$JRE_Facility_Code 3 /* Wrong facility code */
&GLOBAL-DEFINE PRM$JRE_Site_Code 4 /* Wrong site code */
&GLOBAL-DEFINE PRM$JRE_PIN 5 /* Wrong PIN */
&GLOBAL-DEFINE PRM$JRE_Issue_Code 6 /* Wrong issue code */
&GLOBAL-DEFINE PRM$JRE_Lost 7 /* Card flagged as lost */
&GLOBAL-DEFINE PRM$JRE_Disabled 8 /* Card flagged as disabled */
&GLOBAL-DEFINE PRM$JRE_Expired 9 /* Card has expired */
&GLOBAL-DEFINE PRM$JRE_Not_Activated 10 /* Card was not activated */
&GLOBAL-DEFINE PRM$JRE_Not_Downloaded 11 /* Card was not downloaded */
&GLOBAL-DEFINE PRM$JRE_Illegal_Reject_Code 12 /* The apC sent an unsupported
reject code */
&GLOBAL-DEFINE PRM$JRE_Misread 13 /* Card misread */
&GLOBAL-DEFINE PRM$JRE_Tailgate 14 /* Person tailgated */
&GLOBAL-DEFINE PRM$JRE_Passback 15 /* Card was passed back */
&GLOBAL-DEFINE PRM$JRE_Timed_AP 16 /* Timed antipassback violation */
&GLOBAL-DEFINE PRM$JRE_Floor 17 /* Cardholder accessed wrong floor
*/
&GLOBAL-DEFINE PRM$JRE_Linked_Asset 18 /* Linked Asset not valid movement
*/
&GLOBAL-DEFINE PRM$JRE_RSRV1 19 /* Reserved for iStar diagnostic */
&GLOBAL-DEFINE PRM$JRE_RSRV2 20 /* Reserved for iStar diagnostic */
&GLOBAL-DEFINE PRM$JRE_Invalid_Escort 21 /* Invalid Escort access*/
&GLOBAL-DEFINE PRM$JRE_No_Escort 22 /* Escort not present */
&GLOBAL-DEFINE PRM$JRE_Unknown_Command 23 /* Keypad Command failed - Not in
database, record ID = 0*/
&GLOBAL-DEFINE PRM$JRE_Command_Disabled 24 /* Keypad Command failed - Command
exists but is configured to be
disabled*/
&GLOBAL-DEFINE PRM$JRE_Door 25 /* Keypad Command failed - Command
not allowed at this door*/
&GLOBAL-DEFINE PRM$JRE_Permission 26 /* Keypad Command failed - This
cardholder not allowed to issue
command*/
&GLOBAL-DEFINE PRM$JRE_No_Card 27 /* Keypad Command failed - No card
presented (timeout)*/
&GLOBAL-DEFINE PRM$JRE_No_PIN 28 /* Keypad Command failed - No PIN
presented (timeout)*/
&GLOBAL-DEFINE PRM$JRE_Misconfigured 29 /* Keypad Command failed -
Configuration problem prevents
issuance*/
&GLOBAL-DEFINE PRM$JRE_Command_Format 30 /* Keypad Command failed - Format of
command entered through prompts
incorrect*/
&GLOBAL-DEFINE PRM$JRE_Timeout 31 /* Keypad Command failed - Generic
timeout */
&GLOBAL-DEFINE PRM$JRE_KPC_not_enabled 32 /* Keypad Command failed - - Keypad
Commands are not enabled at the
reader*/
&GLOBAL-DEFINE PRM$JRE_RdrRptMisread 33 /* Reader Reported Misread, record
ID = 0 */
&GLOBAL-DEFINE PRM$JRE_Exp_Clearance 34 /* reject - clearance expired */
&GLOBAL-DEFINE PRM$JRE_Act_Clearance 35 /* reject - clearance not activated
*/
&GLOBAL-DEFINE PRM$JRE_Area_Lockout 36 /* reject - area lockout */
&GLOBAL-DEFINE PRM$JRE_Card_Status 37 /* reject - card status field
indicated reject */
&GLOBAL-DEFINE PRM$JRE_Area_Maximum 38 /* reject - area has maximum
personnel */
&GLOBAL-DEFINE PRM$JRE_Area_Minimum 39 /* reject - area has minimum
personnel */
&GLOBAL-DEFINE PRM$JRE_Area_PGroup_Max 40 /* reject - area personnel group max
*/
&GLOBAL-DEFINE PRM$JRE_Area_PGroup_Min 41 /* reject - area personnel group min
*/

```

```

&GLOBAL-DEFINE PRM$JRE_Area_MixPerson          42 /* reject - mixed personnel swiping
card in one area
door */
&GLOBAL-DEFINE PRM$JRE_Area_MixPGroup          43 /* reject - mixed PG personnel
swiping card in one
area door */
&GLOBAL-DEFINE PRM$JRE_Area_NotInPGroup        44 /* reject - person not belonging to
personnel group for area
*/
&GLOBAL-DEFINE PRM$JRE_Area_NoSupervisor        45 /* reject - Supervised cannot enter
because no supervisor */
&GLOBAL-DEFINE PRM$JRE_Area_NoEscort           46 /* reject - Visitor cannot enter
because no escort */
&GLOBAL-DEFINE PRM$JRE_Area_UnattendedSuper    47 /* reject - Supervisor cannot leave
because supervised
personnels would be
unattended */
&GLOBAL-DEFINE PRM$JRE_Area_UnattendedEscort   48 /* reject - Escort cannot leave
since escorted visitors
would be unattended */
&GLOBAL-DEFINE PRM$JRE_Area_CountMismatch      49 /* reject - area count reaches zero,
but still people inside
*/
&GLOBAL-DEFINE PRM$JRE_Area_PGCountMismatch    50 /* reject - area pgroup count
reaches zero, but still
people inside */
&GLOBAL-DEFINE PRM$JRE_Busy                    51 /* reject - one reader of the area
is processing card for
minimum, other readers
will wait */
&GLOBAL-DEFINE PRM$JRE_Carpool_Driver_Mismatch 52 /* reject - the exit card is
different from the
entrance card for
carpool access */
&GLOBAL-DEFINE PRM$JRE_GAPB_Passback_Comm_Failure 53 /* reject - GAPB reject due to comm
failure */
&GLOBAL-DEFINE PRM$JRE_Card_Disabled           54 /* reject - Card is disabled */
&GLOBAL-DEFINE PRM$JRE_Stolen                  55 /* reject - Card is stolen */
&GLOBAL-DEFINE PRM$JRE_Multiple_Card           56 /* reject - Card is not first valid
non-extended card */
&GLOBAL-DEFINE PRM$JRE_Extended_Card           57 /* reject - Card is extended and not
downloaded to panel */
&GLOBAL-DEFINE PRM$JRE_Last_Card_Reject        57 /* highest logging code */
/*

```

States an object can be in that might cause an action.

```

*/
enum ObjectStateCode
{
    PRM$OST_None                = 1,    /* No status */
    PRM$OST_Active               = 2,    /* Object is in active state */
    PRM$OST_Inactive             = 3,    /* Object is in inactive state */
    PRM$OST_Mom_Active           = 4,    /* Object is momentarily activated */
    PRM$OST_On_Line              = 5,    /* Object is on-line */
    PRM$OST_Off_Line             = 6,    /* Object is off-line */
    PRM$OST_Supervision          = 7,    /* Generic supervision error */
    PRM$OST_Grounded_Loop        = 8,    /* Supervision - grounded loop */
    PRM$OST_Shorted_Loop         = 9,    /* Supervision - shorted loop */
    PRM$OST_Open_Loop            = 10,   /* Supervision - open fault */
    PRM$OST_Fault                = 11,   /* Supervision - fault */
    PRM$OST_Locked               = 12,   /* Object is locked */
    PRM$OST_Unlocked             = 13,   /* Object is unlocked */
    PRM$OST_Secure               = 14,   /* Object is secure (no access) */
    PRM$OST_Armed                = 15,   /* Object is armed */
    PRM$OST_Disarmed             = 16,   /* Object is disarmed */
    PRM$OST_Neutral              = 17,   /* Object state is undefined */
    PRM$OST_Active_in_TimeSpec    = 18,   /* Active w/in timespec boundary */
    PRM$OST_Active_Outside_TimeSpec = 19, /* Active outside timespec boundary */
    PRM$OST_ADA_Unlocked         = 20,   /* Object is ADA unlocked */
    PRM$OST_Door_Forced          = 26,   /* Door forced */
    PRM$OST_Door_Held            = 27,   /* Door held */
    PRM$OST_Admit                = 28,   /* Card admit */
    PRM$OST_Reject               = 29,   /* Card reject */
    PRM$OST_Visitor_Admit        = 30,   /* Visitor admit */

```

```

PRM$OST_Visitor_Reject           = 31, /* Visitor reject */
PRM$OST_Noticed_Admit           = 32, /* Noticed card admit */
PRM$OST_Noticed_Reject         = 33, /* Noticed card reject */
PRM$OST_Duress                  = 35, /* Card duress */
PRM$OST_Tamper                  = 37, /* Tamper */
PRM$OST_Power_failure           = 38, /* Power failure */
PRM$OST_Communications_failure = 39, /* Communications failure */
PRM$OST_Communications_restored = 40, /* Communications restored */
PRM$OST_Power_restored          = 41, /* Power restored */
PRM$OST_Tamper_cleared          = 42, /* Tamper cleared */
PRM$OST_Door_closed             = 43, /* Door closed */
PRM$OST_Door_open               = 44, /* Door open */
PRM$OST_Supervision_Cleared     = 45, /* Generic supervision error */
PRM$OST_Grounded_Loop_Cleared   = 46, /* Supervision - grounded loop */
PRM$OST_Shorted_Loop_Cleared    = 47, /* Supervision - shorted loop */
PRM$OST_Open_Loop_Cleared       = 48, /* Supervision - open fault */
PRM$OST_Fault_Cleared           = 49, /* Supervision - fault */
PRM$OST_Acknowledge             = 50, /* Acknowledge */
PRM$OST_Mom_Unlock              = 51, /* Object is momentarily unlocked */
PRM$OST_Controlled_Access       = 57, /* Elevator/floor is controlled
access */
PRM$OST_Uncontrolled_Access     = 58, /* Elevator/floor uncontrolled
access */
PRM$OST_Connection_failure      = 61, /* Dialup connection error */
PRM$OST_Asset_Reject            = 71, /* Card denied for asset */
PRM$OST_Asset_Checkin           = 81, /* Overdue asset got checked in */
PRM$OST_Printer_Buffer_Overflow = 82, /* Activity printer buffer overflow
*/
PRM$OST_Printer_Abnormal        = 83, /* Activity printer in abnormal
state */
PRM$OST_Printer_Paper_Jam       = 84, /* Activity printer paper jam */
PRM$OST_Printer_Out_Of_Paper    = 85, /* Activity printer out of paper */
PRM$OST_Printer_Offline         = 87, /* Activity printer general error:
comm failure.*/
PRM$OST_Printer_Normal          = 88, /* Activity printer is normal */
PRM$OST_Printer_Buffer_Overflow_cleared = 89, /* Activity printer buffer overflow
cleared */
PRM$OST_Printer_Paper_Jam_cleared = 90, /* Activity printer paper jam
cleared */
PRM$OST_Printer_Out_Of_Paper_cleared = 91, /* Activity printer out of paper
cleared */
PRM$OST_Printer_Online          = 92, /* Activity printer online
physically */
PRM$OST_Printer_General_Error_cleared = 93, /* Activity printer general error:
comm failure */
PRM$OST_PIN_Required            = 94, /* Reader requires PIN */
PRM$OST_PIN_Disabled            = 95, /* Reader PIN disabled */
PRM$OST_Printer_Power_Off       = 96, /* Activity printer power off */
PRM$OST_Printer_Power_On        = 97, /* Activity printer power on */
PRM$OST_Page_Failed             = 98, /* page failed */
PRM$OST_Email_Failed            = 99, /* e-mail failed */
PRM$OST_Control_Zone_Mode_Secure = 100, /* intrusion zone mode set to secure
*/
PRM$OST_Control_Zone_Mode_Access = 101, /* intrusion zone mode set to access
*/
PRM$OST_Control_Zone_Access_Input = 102, /* intrusion zone disarming input */
PRM$OST_Control_Zone_Secure_Input = 103, /* intrusion zone arming input */
PRM$OST_Control_Zone_Access_Tamper = 104, /* intrusion zone disarming tamper
*/
PRM$OST_Control_Zone_Secure_Tamper = 105, /* intrusion zone arming tamper */
PRM$OST_Control_Zone_Access_Output = 106, /* intrusion zone disarming output
*/
PRM$OST_Control_Zone_Secure_Output = 107, /* intrusion zone arming output */
PRM$OST_Control_Zone_Violated_Output = 108, /* intrusion zone violated output */
PRM$OST_Control_Zone_Input_Off_Normal = 109, /* intrusion zone input offnormal */
PRM$OST_Control_Zone_Input_Normal = 110, /* intrusion zone input normal */
PRM$OST_Control_Zone_Door_Open   = 111, /* intrusion zone door open */
PRM$OST_Control_Zone_Door_Closed = 112, /* intrusion zone door open */
PRM$OST_Control_Zone_General_Input = 113, /* intrusion zone general input */
PRM$OST_Primary_Comm_Method_Fail = 114, /* cluster primary comm method
failed (controller) */
PRM$OST_Secondary_Comm_Method_Fail = 115, /* cluster secondary comm method
failed (controller) */

```

```

PRM$OST_Control_Zone_State_Violated           = 116, /* intrusion zone state set to
                                                violated */
PRM$OST_Control_Zone_NotSecure                = 117, /* intrusion zone not secure */
PRM$OST_Control_Zone_Access_Secure_Input      = 118, /* intrusion zone arming/diarming
mode input */
PRM$OST_Primary_Comm_Method_Fail_Host         = 119, /* cluster primary comm method
failed (host) */
PRM$OST_Secondary_Comm_Test_Restored          = 120, /* secondary comm test finally
succeeded after failure */
PRM$OST_Slave_Master_Comm_Fail                = 121, /* comm failed to master */
PRM$OST_Secondary_Comm_Test_Fail              = 122, /* secondary comm method test failed
*/
PRM$OST_Low_Battery                           = 123, /* low battery on controller */
PRM$OST_Primary_Comm_Method_Restored_Host     = 124, /* cluster primary comm method
restored(host) */
PRM$OST_Secondary_Comm_Method_Restored_Host   = 125, /* cluster secondary comm method
restored */
PRM$OST_Cluster_Split                          = 126, /* cluster is communicating via both
methods */
PRM$OST_Cluster_Not_Split                     = 127, /* cluster is communicating via
primary OR secondary, or none */
PRM$OST_Secondary_Comm_Method_Fail_Host       = 128, /* cluster secondary comm method
failed(host)*/
PRM$OST_Panel_Full                            = 129, /* Panel becomes full */
PRM$OST_Panel_Nearly_Full                     = 130, /* Panel becomes nearly full (reaches
given percentage of capacity) */
PRM$OST_Panel_Not_Full                       = 131, /* Panel becomes not full after it
was full*/
PRM$OST_Panel_Not_Nearly_Full                 = 132, /* Panel becomes not nearly full
after it was nearly full */
PRM$OST_Admit_Reject_CCTV                    = 133, /* Admit/reject cause CCTV set alarm
action */
PRM$OST_Shunt_Expire_Warning                  = 134, /* Shunt expiration output is
activated */
PRM$OST_TourStop_Input                        = 135, /* TourStopObject relation to input
*/
PRM$OST_Tour_End_Early                       = 136, /* Tour end early event state */
PRM$OST_Tour_End_Late                        = 137, /* Tour end late event state */
PRM$OST_TourStop_Reached_OutOfSeq            = 138, /* Tourstop reached out of sequence
*/
PRM$OST_TourStop_Reached_Early                = 139, /* Tourstop reached early event
state */
PRM$OST_TourStop_Reached_Late                 = 140, /* Tourstop reached late event state
*/
PRM$OST_RadReceiver_Battery_Fail              = 141, /* Radionics Receiver: Event to
activate on battery failure */
PRM$OST_RadReceiver_Battery_Restored          = 142, /* Radionics Receiver: Event to
activate on battery restored */
PRM$OST_Door_Position_Sensor                  = 143, /* Door position sensor */
PRM$OST_Lock_Status_Sensor                    = 144, /* Lock status sensor */
PRM$OST_Set_Event                            = 145,
PRM$OST_Reset_Event                           = 146,
PRM$OST_Last_State                           = 146
};

```

```

/*
  Action codes. Defines all possible actions that the state change of one object could
  induce upon another object
*/
&GLOBAL-DEFINE PRM$ACT_none 1
&GLOBAL-DEFINE PRM$ACT_Lock_Door 2
&GLOBAL-DEFINE PRM$ACT_Unlock_Door 3
&GLOBAL-DEFINE PRM$ACT_Mom_Unlock_Door 4
&GLOBAL-DEFINE PRM$ACT_Secure_Door 5
&GLOBAL-DEFINE PRM$ACT_Activate_Event 6
&GLOBAL-DEFINE PRM$ACT_Deactivate_Event 7
&GLOBAL-DEFINE PRM$ACT_Arm_Event 8
&GLOBAL-DEFINE PRM$ACT_Disarm_Event 9
&GLOBAL-DEFINE PRM$ACT_Arm_Input 10
&GLOBAL-DEFINE PRM$ACT_Disarm_Input 11
&GLOBAL-DEFINE PRM$ACT_Activate_Output 12
&GLOBAL-DEFINE PRM$ACT_Deactivate_Output 13
&GLOBAL-DEFINE PRM$ACT_Pulse_Output 14
&GLOBAL-DEFINE PRM$ACT_Acknowledge 15
&GLOBAL-DEFINE PRM$ACT_Reset_Actions 16
&GLOBAL-DEFINE PRM$ACT_Pulse_Event 17
&GLOBAL-DEFINE PRM$ACT_Grace_Person 18
&GLOBAL-DEFINE PRM$ACT_Control_Access 19
&GLOBAL-DEFINE PRM$ACT_Uncontrol_Access 20
&GLOBAL-DEFINE PRM$ACT_Control_CCTV 21
&GLOBAL-DEFINE PRM$ACT_Apc_Connect 22
&GLOBAL-DEFINE PRM$ACT_Apc_Disconnect 23
&GLOBAL-DEFINE PRM$ACT_Uncontrol_CCTV 24
&GLOBAL-DEFINE PRM$ACT_Initiate_Roll_Call_Report 25
&GLOBAL-DEFINE PRM$ACT_Grace_All_Cards 26
&GLOBAL-DEFINE PRM$ACT_Require_PIN 27
&GLOBAL-DEFINE PRM$ACT_Disable_PIN 28
&GLOBAL-DEFINE PRM$ACT_Send_Page 29
&GLOBAL-DEFINE PRM$ACT_Send_Email 30
&GLOBAL-DEFINE PRM$ACT_Flash_Download 31
&GLOBAL-DEFINE PRM$ACT_Cancel_Flash 32
&GLOBAL-DEFINE PRM$ACT_ControlZone_Access 33
&GLOBAL-DEFINE PRM$ACT_ControlZone_Secure 34
&GLOBAL-DEFINE PRM$ACT_ControlZone_ForceSecure 35
&GLOBAL-DEFINE PRM$ACT_ControlZone_GetControlZoneStatus 36
&GLOBAL-DEFINE PRM$ACT_ControlZone_VerifySecure 37
&GLOBAL-DEFINE PRM$ACT_ControlZone_StopVerifySecure 38
&GLOBAL-DEFINE PRM$ACT_Connect_Controller 39
&GLOBAL-DEFINE PRM$ACT_Disconnect_Controller 40
&GLOBAL-DEFINE PRM$ACT_Test_Primary_Connection 41
&GLOBAL-DEFINE PRM$ACT_Test_Secondary_Connection 42
&GLOBAL-DEFINE PRM$ACT_Deactivate_Event_Pending_Min_Act_time 43
&GLOBAL-DEFINE PRM$ACT_Deactivate_Event_Min_Act_time_Cancelled 44
&GLOBAL-DEFINE PRM$ACT_Activate_Event_Pending_Act_Delay 45
&GLOBAL-DEFINE PRM$ACT_Activate_Event_Act_Delay_Cancelled 46
&GLOBAL-DEFINE PRM$ACT_CCTV_Set_Title_Action 47
&GLOBAL-DEFINE PRM$ACT_Watchtour_Start 48
&GLOBAL-DEFINE PRM$ACT_Watchtour_Cancel 49
&GLOBAL-DEFINE PRM$ACT_Watchtour_Suspend 50
&GLOBAL-DEFINE PRM$ACT_Watchtour_Resume 51
&GLOBAL-DEFINE PRM$ACT_WatchFlash_Download 52
&GLOBAL-DEFINE PRM$ACT_Watch_DspFileleter_Download 53
&GLOBAL-DEFINE PRM$ACT_Activate_Selected_WatchFlash 54
&GLOBAL-DEFINE PRM$ACT_Request_FileUpload 55
&GLOBAL-DEFINE PRM$ACT_Request_FileUpload_Timed 56
&GLOBAL-DEFINE PRM$ACT_Control_NetVideo 57
&GLOBAL-DEFINE PRM$ACT_Set_Event 58
&GLOBAL-DEFINE PRM$ACT_Reset_Event 59
&GLOBAL-DEFINE PRM$ACT_Last_Action 59

```

```

/*
Journal manual action code strings. Defines all journal manual action codes. Used
in the manual action code field of the PRM$JMC_Manual_Action.
*/
&GLOBAL-DEFINE PRM$JMA_Scheduled 1 /* A manual action has been
scheduled */
&GLOBAL-DEFINE PRM$JMA_Activated 2 /* A manual action has become
active */
&GLOBAL-DEFINE PRM$JMA_Canceled 3 /* A manual action has been
cancelled */
&GLOBAL-DEFINE PRM$JMA_Deactivated 4 /* A manual action has completed
*/
&GLOBAL-DEFINE PRM$JMA_Momentary 5 /* A momentary action has occurred
*/
&GLOBAL-DEFINE PRM$JMA_Acknowledge 6 /* An event Acknowledgment has
occurred */
&GLOBAL-DEFINE PRM$JMA_Reset_Actions 7 /* An event reset actions button
was clicked */
&GLOBAL-DEFINE PRM$JMA_Reject_Actions 8 /* A manual action has rejected */
&GLOBAL-DEFINE PRM$JMA_Reject_Cancel_Actions 9 /* A manual cancel action was
rejected */
&GLOBAL-DEFINE PRM$JMA_Last_Manual_Action 9 /* Should always equal the highest
code */

*/
/*
Directionality of readers
*/
/* enum JournalReaderDirectionality */
/* { */
&GLOBAL-DEFINE PRM$NEUTRAL_READ 0
&GLOBAL-DEFINE PRM$IN_READ 1
&GLOBAL-DEFINE PRM$out_READ 2
&GLOBAL-DEFINE PRM$UNSPECIFIED 3
/*

*/
/*
Journal asset movement codes.
List of possible asset movement codes.
*/
/* enum JournalAssetMovementCode */
/* { */
&GLOBAL-DEFINE PRM$JAM_Authorized_Accompanied 0 /* Authorized with cardholder */
&GLOBAL-DEFINE PRM$JAM_Unauthorized_Accompanied 1 /* Unauthorized with cardholder */
&GLOBAL-DEFINE PRM$JAM_Attempted_Accompanied 2 /* Attempted with cardholder */
&GLOBAL-DEFINE PRM$JAM_Authorized_Unaccompanied 3 /* Authorized with cardholder */
&GLOBAL-DEFINE PRM$JAM_Unauthorized_Unaccompanied 4 /* Unauthorized with cardholder */
&GLOBAL-DEFINE PRM$JAM_Attempted_Unaccompanied 5 /* Attempted with cardholder */
&GLOBAL-DEFINE PRM$JAM_Last_Movement_Code 5 /* Should always equal the highest
code */
/* }; */

*/
/*
Journal system activity codes. Defines all journal system activity codes. Used in
the activity code field of the PRM$JMC_System_Activity message.
*/
&GLOBAL-DEFINE PRM$JSM_System_Startup 1 /* System startup */
&GLOBAL-DEFINE PRM$JSM_System_Shutdown 2 /* System shutdown */
&GLOBAL-DEFINE PRM$JSM_Journal_File_Changed 3 /* New journal file */
&GLOBAL-DEFINE PRM$JSM_System_Backup_Start 4 /* System backup initiated */
&GLOBAL-DEFINE PRM$JSM_Controller_Denied 5 /* Rejected iStatcontroller connection
*/
&GLOBAL-DEFINE PRM$JSM_ER_DBReset_Requested 6 /* Enhanced Reports Database reset
requested by Admin */
&GLOBAL-DEFINE PRM$JSM_ER_DBReset_Completed 7 /* Enhanced Reports Database reset
&GLOBAL-DEFINE PRM$JSM_ER_DBReset_Failed 8 /* Enhanced Reports Database reset
failed */
&GLOBAL-DEFINE PRM$JSM_ER_DBReset_Too_Big 9 /* Enhanced Reports Database reset
rejected because database is too big
*/
&GLOBAL-DEFINE PRM$JSM_Last_System_Message 9 /* Should always equal the highest code
*/

*/
/*

```

Journal system error codes. Defines all journal system error codes. Used in the error code field of the PRM\$JMC_System_Error message.

```

*/
&GLOBAL-DEFINE PRM$JSE_Disk_error 1 /* Disk error */
&GLOBAL-DEFINE PRM$JSE_Database_error 2 /* Database error */
&GLOBAL-DEFINE PRM$JSE_API_Call_Failed 3 /* API call error */
&GLOBAL-DEFINE PRM$JSE_Thread_Init_Failure 4 /* Thread init failed */
&GLOBAL-DEFINE PRM$JSE_Using_Mouse_Port 5 /* Attempt to use mouse port */
&GLOBAL-DEFINE PRM$JSE_Error_While_Allocating_Port 6 /* Error while allocating port */
&GLOBAL-DEFINE PRM$JSE_Disk_Space_Low 7 /* Running out of disk space */
&GLOBAL-DEFINE PRM$JSE_Site_Expired 8 /* Site Expired */
&GLOBAL-DEFINE PRM$JSE_Site_Will_Expire 9 /* Site Will Expire */
&GLOBAL-DEFINE PRM$JSE_SSA_Expired 10 /* Software Support Agreement expired
*/
&GLOBAL-DEFINE PRM$JSE_SSA_Will_Expire 11 /* Software Support Agreement will
expire */
&GLOBAL-DEFINE PRM$JSE_Badging_Expired 12 /* Badging expired */
&GLOBAL-DEFINE PRM$JSE_Badging_Will_Expire 13 /* Badging will expire */
&GLOBAL-DEFINE PRM$JSE_Invalid_Sentinel 14 /* Invalid sentinel */
&GLOBAL-DEFINE PRM$JSE_JSE_Unknown_Panel 15 /* Received call from unknown panel */
&GLOBAL-DEFINE PRM$JSE_NTEventLogError 16 /* NT Event Log was full, so a message
was lost */
&GLOBAL-DEFINE PRM$JSE_Asset_Tracking_Will_Expire 17 /* Asset tracking will expire in x days
*/
&GLOBAL-DEFINE PRM$JSE_Asset_Tracking_Expired 18 /* Asset tracking expired */
&GLOBAL-DEFINE PRM$JSE_Paging_Will_Expire 19 /* Paging will expire in x days */
&GLOBAL-DEFINE PRM$JSE_Paging_Expired 20 /* Paging expired */
&GLOBAL-DEFINE PRM$JSE_ER_Connect_Problem 21 /* Enhanced reports connect problem */
&GLOBAL-DEFINE PRM$JSE_ER_Bad_Message 22 /* Enhanced reports got bad message -
this should nothappen */
&GLOBAL-DEFINE PRM$JSE_ER_Missing_Journal_Vol 23 /* Enhanced reports missing journal
volume */
&GLOBAL-DEFINE PRM$JSE_ER_replay_problem 24 /* cc_replay connect problem during
journal export */
&GLOBAL-DEFINE PRM$JSE_Last_System_Error 24 /* Should always equal the highest code
*/
*/

```

Journal device activity codes. Defines all journal device activity codes. Used in the activity code field of the PRM\$JMC_Device_Activity message.

```

*/
&GLOBAL-DEFINE PRM$JDM_Memory_erased 1 /* Unit memory erased */
&GLOBAL-DEFINE PRM$JDM_Hardware_Reset 2 /* Hardware panel reset */
&GLOBAL-DEFINE PRM$JDM_Power_Recycle 3 /* Unit power recycled */
&GLOBAL-DEFINE PRM$JDM_Download_Started 4 /* Download started */
&GLOBAL-DEFINE PRM$JDM_Download_Completed 5 /* Download completed */
&GLOBAL-DEFINE PRM$JDM_Host_Init_Connection_Started 6 /* Host initiated connection
started */
&GLOBAL-DEFINE PRM$JDM_Host_Init_Connection_Completed 7 /* Host initiated connection
completed */
&GLOBAL-DEFINE PRM$JDM_Panel_Init_Connection_Started 8 /* Panel initiated connection
started */
&GLOBAL-DEFINE PRM$JDM_Panel_Init_Connection_Completed 9 /* Panel initiated connection
completed */
&GLOBAL-DEFINE PRM$JDM_Flash_Started 10 /* Firmware download started
*/
&GLOBAL-DEFINE PRM$JDM_Flash_Completed 11 /* Firmware download completed
*/
&GLOBAL-DEFINE PRM$JDM_Email_Sent 12 /* Email send */
&GLOBAL-DEFINE PRM$JDM_Page_Sent 13 /* Page sent */
&GLOBAL-DEFINE PRM$JDM_Grace_All 14 /* Grace All/Grace Partition
*/
&GLOBAL-DEFINE PRM$JDM_Grace_Card 15 /* Grace a single card */
&GLOBAL-DEFINE PRM$JDM_Download_UnitIsFull 16 /* Full Unit is preventing
card download */
&GLOBAL-DEFINE PRM$JDM_iSTAR_Dialup_Connected 17
&GLOBAL-DEFINE PRM$JDM_iSTAR_Dialup_Started 18
&GLOBAL-DEFINE PRM$JDM_RAD_BUSY_SECOND 19
&GLOBAL-DEFINE PRM$JDM_RAD_COMPUTER_ERROR 20
&GLOBAL-DEFINE PRM$JDM_RAD_COMPUTER_RESTORED 21
&GLOBAL-DEFINE PRM$JDM_RAD_PHONE_LINE_FAIL 22
&GLOBAL-DEFINE PRM$JDM_RAD_PHONE_LINE_RESTORED 23
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_CLOSE 24

```

C•CURE 800/8000 ODBC Configuration Guide

```

&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_OPEN 25
&GLOBAL-DEFINE PRM$JDM_RAD_UNKNOWN_MSG 26 /* receiver unknown msg */
&GLOBAL-DEFINE PRM$JDM_RAD_CRC_ERROR 27
&GLOBAL-DEFINE PRM$JDM_RAD_RECEIVER_NUM_WRONG 28
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_CLOSE_ZN 29
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_OPEN_ZN 30
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_CLOSE_ID 31
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_OPEN_ID 32
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_CLOSE_AREA 33
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_OPEN_AREA 34
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_CLOSE_AREA_ID 35
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_OPEN_AREA_ID 36
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_UNKNOWN_MSG 37 /* receiver and account
unknown msg */
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM 38
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_RESTORE 39
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_ZONE 40
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_RESTORE_ZONE 41
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_AREA 42
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_RESTORE_AREA 43
&GLOBAL-DEFINE PRM$JDM_RAD_LINE_CARD_TROUBLE 44
&GLOBAL-DEFINE PRM$JDM_RAD_LINE_CARD_RESTORE 45
&GLOBAL-DEFINE PRM$JDM_RAD_PRINTER_TROUBLE 46
&GLOBAL-DEFINE PRM$JDM_RAD_PRINTER_RESTORE 47
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_DIAGNOSTIC 48
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_DIAGNOSTIC_ZONE 49
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_BATTERY_FAIL 50
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_BATTERY_RESTORE 51
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_AC_FAIL 52
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_AC_RESTORE 53
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_REBOOT 54
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_POINT_BUS_FAIL 55
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_POINT_BUS_RESTORE 56
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_SDI_FAIL 57
&GLOBAL-DEFINE PRM$JDM_RAD_ACCOUNT_SDI_RESTORE 58
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_POINT 59
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_RESTORE_POINT 60
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_AREA 61
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_RESTORE_AREA 62
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_TROUBLE 63
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_TROUBLE_POINT 64
&GLOBAL-DEFINE PRM$JDM_RAD_ALARM_TROUBLE_AREA_POINT 65
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_TROUBLE 66
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_TROUBLE_POINT 67
&GLOBAL-DEFINE PRM$JDM_RAD_FIRE_ALARM_TROUBLE_AREA_POINT 68
&GLOBAL-DEFINE PRM$JDM_RAD_PRINTER_TEST 69
&GLOBAL-DEFINE PRM$JDM_RAD_PRINTER_ONLINE 70
&GLOBAL-DEFINE PRM$JDM_RAD_PRINTER_OFFLINE 71
&GLOBAL-DEFINE PRM$JDM_RAD_CANCEL_ALARM_ID 72
&GLOBAL-DEFINE PRM$JDM_RAD_CANCEL_ALARM_AREA_ID 73
&GLOBAL-DEFINE PRM$JDM_RAD_CANCEL_FIRE_ALARM_AREA_ID 74
&GLOBAL-DEFINE PRM$JDM_WatchFlash_Download_Started 75 /* CCure Watch Firmware
download started */
&GLOBAL-DEFINE PRM$JDM_WatchFlash_Download_Completed 76 /* CCure Watch Firmware
download completed */
&GLOBAL-DEFINE PRM$JDM_WatchFlash_Swapped 77 /* CCure Watch Firmware
swapped */
&GLOBAL-DEFINE PRM$JDM_WatchFlash_Upload_Started 78 /* CCure Watch Firmware
upload started */
&GLOBAL-DEFINE PRM$JDM_WatchFlash_Upload_Completed 79 /* CCure Watch Firmware
upload completed */
&GLOBAL-DEFINE PRM$JDM_Watch_Flash_Loading_Canceled 80
&GLOBAL-DEFINE PRM$JDM_NetVideo_Server_Comm_Error 81
&GLOBAL-DEFINE PRM$JDM_NetVideo_Server_Comm_Restored 82
&GLOBAL-DEFINE PRM$JDM_NetVideo_Status_Retry 83
&GLOBAL-DEFINE PRM$JDM_NetVideo_Server_Pipe_Timeout 84
&GLOBAL-DEFINE PRM$JDM_NetVideo_Server_Error 85
&GLOBAL-DEFINE PRM$JDM_NetVideo_Camera_Error 86
&GLOBAL-DEFINE PRM$JDM_NetVideo_Action_Error 87
&GLOBAL-DEFINE PRM$JDM_NetVideo_Server_Comm_Success 88
&GLOBAL-DEFINE PRM$JDM_BID_Receiver_JnlMsg 89
&GLOBAL-DEFINE PRM$JDM_BID_Action_JnlMsg1 90

```

&GLOBAL-DEFINE	PRM\$JDM_BID_Action_JnlMsg2	91
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_JnlMsg3	92
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_JnlMsg4	93
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_JnlMsg5	94
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_JnlMsg6	95
&GLOBAL-DEFINE	PRM\$JDM_BID_Receiver_JnlMsg1	96
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_No_Command	97
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_Empty_Command	98
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_Set_Command_Err	99
&GLOBAL-DEFINE	PRM\$JDM_BID_Action_Device_Comm_Err	100
&GLOBAL-DEFINE	PRM\$JDM_Watch>Loading_Canceled	101
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Server_Online	102
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Server_Offline	103
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Server_Initial	104
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Server_Initial_Err	105
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Process_Shutdown	106
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Connection_Error	107
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Invalid_Vendor_Info	108
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_IP_Error	109
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Status_Msg_Error	110
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Status_Msg_Timeout	111
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Status_Msg_Unknown	112
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Status_Error	113
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Invalid_Status	114
&GLOBAL-DEFINE	PRM\$JDM_NetVideo_Unknown_Status	115
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Config_Error	116
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Camera_Num_Error	117
&GLOBAL-DEFINE	PRM\$JDM_Handling_Vendor_Info_Error	118
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Recording	119
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Connected	120
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Disconnected	121
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Stopped	122
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Initial	123
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Status_Invalid	124
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Status_Unknown	125
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Without_Server	126
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Number_Error	127
&GLOBAL-DEFINE	PRM\$JDM_Video_Action_Without_Server	128
&GLOBAL-DEFINE	PRM\$JDM_Video_Action_Server_Offline	129
&GLOBAL-DEFINE	PRM\$JDM_Video_Action_Without_Camera	130
&GLOBAL-DEFINE	PRM\$JDM_Video_Action_Camera_Offline	131
&GLOBAL-DEFINE	PRM\$JDM_BID_Receiver_ReInit	132
&GLOBAL-DEFINE	PRM\$JDM_Video_Time_Sync	133
&GLOBAL-DEFINE	PRM\$JDM_Video_Time_Sync_Failed	134
&GLOBAL-DEFINE	PRM\$JDM_Device_Flash_File_DL_Complete	135
&GLOBAL-DEFINE	PRM\$JDM_Device_Flash_Complete	136
&GLOBAL-DEFINE	PRM\$JDM_Host_Demuster_Start	137
&GLOBAL-DEFINE	PRM\$JDM_Host_Demuster_Complete	138
&GLOBAL-DEFINE	PRM\$JDM_Hardware_Demuster_OneArea	139
&GLOBAL-DEFINE	PRM\$JDM_Hardware_Demuster_All	140
&GLOBAL-DEFINE	PRM\$JDM_BID_Gateway_Message1	141
&GLOBAL-DEFINE	PRM\$JDM_BID_Input_LowBattery	142
&GLOBAL-DEFINE	PRM\$JDM_BID_Input_LowBattery_Cleared	143
&GLOBAL-DEFINE	PRM\$JDM_BID_Input_Tamper	144
&GLOBAL-DEFINE	PRM\$JDM_BID_Input_Tamper_Cleared	145
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Alarm_Active	146
&GLOBAL-DEFINE	PRM\$JDM_Video_Server_Alarm_Inactive	147
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Alarm_Active	149
&GLOBAL-DEFINE	PRM\$JDM_Video_Camera_Alarm_Inactive	150
&GLOBAL-DEFINE	PRM\$JDM_BID_Gateway_Message2	151
&GLOBAL-DEFINE	PRM\$JDM_Download_DueTo_PanelReset	152
&GLOBAL-DEFINE	PRM\$JDM_Download_DueTo_NewMaster	153
&GLOBAL-DEFINE	PRM\$JDM_Download_DueTo_Online	154
&GLOBAL-DEFINE	PRM\$JDM_Video_Display_Live	155

```

&GLOBAL-DEFINE PRM$JDM_Fast_Card_Download_Started          156 /* Fast Download started
*/
&GLOBAL-DEFINE PRM$JDM_Fast_Card_Download_Completed        157 /* Fast Download
completed */

&GLOBAL-DEFINE PRM$JDM_Fast_Flash_Started                  158 /* Fast Firmware
download started */
&GLOBAL-DEFINE PRM$JDM_Fast_Flash_Completed                159 /* Fast Firmware
download completed */

&GLOBAL-DEFINE PRM$JDM_Fast_Card_Download_DueTo_NewMaster  160
&GLOBAL-DEFINE PRM$JDM_Fast_Card_Download_DueTo_Online    161

&GLOBAL-DEFINE PRM$JDM_IStar_Database_Backup_Started      162
&GLOBAL-DEFINE PRM$JDM_IStar_Database_Backup_Ended        163

&GLOBAL-DEFINE PRM$JDM_AreaLockout_Grace                  164
&GLOBAL-DEFINE PRM$JDM_GAPB_Grace_Person_Acked            165
&GLOBAL-DEFINE PRM$JDM_GAPB_Grace_All_Acked               166
&GLOBAL-DEFINE PRM$JDM_GAPB_ALockout_Grace_Person_Acked   167
&GLOBAL-DEFINE PRM$JDM_GAPB_Grace_Person_UnAcked          168
&GLOBAL-DEFINE PRM$JDM_GAPB_Grace_All_UnAcked             169
&GLOBAL-DEFINE PRM$JDM_GAPB_ALockout_Grace_Person_UnAcked 170
&GLOBAL-DEFINE PRM$JDM_GAPB_Reset_Owner_Acked            171
&GLOBAL-DEFINE PRM$JDM_GAPB_Reset_Owner_UnAcked           172
&GLOBAL-DEFINE PRM$JDM_GAPB_Grace_System                  173
&GLOBAL-DEFINE PRM$JDM_GAPB_Reset_System                  174
&GLOBAL-DEFINE PRM$JDM_Video_Display_Live_Tour            175
&GLOBAL-DEFINE PRM$JDM_Last_Device_Message                175 /*Should always equal
the highest code */

/* }; */

Journal communications activity codes. Defines all journal comm activity codes. Used
in the activity code field of the PRM$JCM_Comm_Activity message.
*/
/* enum JournalCommunicationsActivityCode */
/* { */
&GLOBAL-DEFINE PRM$JCM_Dummy 1 /* !!Not yet used!! */
&GLOBAL-DEFINE PRM$JCM_Last_Comm_Message 1 /* Should always equal the
highest code */

/* */
/*
Journal communications error codes. Defines all journal comm error codes. Used
in the activity code field of the PRM$JCE_Comm_Error message.
*/
/* enum JournalCommunicationsErrorCode */
/* { */
&GLOBAL-DEFINE PRM$JCE_Dummy 1 /* !!Not yet used!! */
&GLOBAL-DEFINE PRM$JCE_Last_Comm_Error 1 /* Should always equal the
highest code */

/* */
/*
Journal asset admit codes. Defines all journal asset admit codes. Used in the
admit code field of the asset movement messages.
List of possible flags indicated in the upper 3 bytes of access code. */
/* enum JournalAssetAdmitCode */
/* { */
&GLOBAL-DEFINE PRM$JAA_InDirection 0 /* Direction of reader was In */
&GLOBAL-DEFINE PRM$JAA_OutDirection 1 /* Direction of reader was Out*/
&GLOBAL-DEFINE PRM$JAA_Unmoved 2 /* asset not moved */
&GLOBAL-DEFINE PRM$JAA_Unspecified 3 /* Unspecified read */
&GLOBAL-DEFINE PRM$JAA_CheckIn 4 /* Checked In */
&GLOBAL-DEFINE PRM$JAA_CheckOut 5 /* Checked Out */
&GLOBAL-DEFINE PRM$JAA_Accompanied 6 /* Accompanied with access */
&GLOBAL-DEFINE PRM$JAA_Stationary 7 /* Stationary Asset */
&GLOBAL-DEFINE PRM$JAA_Noticed 8 /* Noticed Asset */
&GLOBAL-DEFINE PRM$JAA_Temp1 9 /* temp 1 */
&GLOBAL-DEFINE PRM$JAA_Temp2 10 /* temp 2 */

&GLOBAL-DEFINE PRM$JAA_LastAdmitCode 10 /* Should always equal the highest code */
&GLOBAL-DEFINE PRM$JAA_NumAdmitCodes 11 /* Need this because above is base zero...
*/

```

```

/* Following are the bit masks for the above asset admit codes. */

/* enum JournalAssetAdmitCodeBitMask */
/* { */
&GLOBAL-DEFINE PRM$JAA_B_InDirection 1 /* Direction of reader was In */
&GLOBAL-DEFINE PRM$JAA_B_OutDirection 2 /* Direction of reader was Out */
&GLOBAL-DEFINE PRM$JAA_B_Unmoved 4 /* asset not moved */
&GLOBAL-DEFINE PRM$JAA_B_Unspecified 8 /* Unspecified read */
&GLOBAL-DEFINE PRM$JAA_B_CheckIn 16 /* Checked In */
&GLOBAL-DEFINE PRM$JAA_B_CheckOut 32 /* Checked Out */
&GLOBAL-DEFINE PRM$JAA_B_Accompanied 64 /* Accompanied with access */
&GLOBAL-DEFINE PRM$JAA_B_Stationary 128 /* Stationary Asset */
&GLOBAL-DEFINE PRM$JAA_B_Noticed 256 /* Noticed Asset */
&GLOBAL-DEFINE PRM$JAA_B_Temp1 512 /* temp 1 */
&GLOBAL-DEFINE PRM$JAA_B_Temp2 1024 /* temp 2 */
/*

```

Journal Activity Codes

```

* enum JournalAssetActivityCode */
/* { */
&GLOBAL-DEFINE PRM$JAT_Overdue 1 /* Asset overdue */
&GLOBAL-DEFINE PRM$JAT_Checkout 2 /* Asset checked out */
&GLOBAL-DEFINE PRM$JAT_Checkin 3 /* Asset checked in */
&GLOBAL-DEFINE PRM$JAT_Last_Asset_Activity_Code 3 /* Should always equal the highest code */
/*
*/ };
*/

```

Journal asset access codes.

List of reasons why a movement could be unauthorized/authorized.

```

*/
enum JournalAssetRejectCode
{
    PRM$JRC_Authorized = 0, /* No linking */
    PRM$JRC_Authorized_With_Owner = 1, /* Movement with owner */
    PRM$JRC_Authorized_With_Group = 2, /* Movement with group member */
    PRM$JRC_Authorized_With_Other = 3, /* Movement with everyone */
    PRM$JRC_Authorized_With_Admin = 4, /* Movement with admin */
    PRM$JRC_Unauthorized_No_Card = 5, /* No accompanying card holder */
    PRM$JRC_Unauthorized_Cardholder = 6, /* Cardholder did not have privileges */
    PRM$JRC_Unauthorized_Time_Spec = 7, /* Movement outside permitted times */
    PRM$JRC_Unauthorized_Stationary = 8, /* Asset type not suppose to move */
    PRM$JRC_Unauthorized_Unknown = 9, /* Asset not in database */
    PRM$JRC_Unauthorized_Clearance = 10, /* Cardholder did not have door clearance */
    PRM$JRC_Unauthorized_Card_Unknown = 11 /* Card not on file */
};

```