RMIS View/Print Document Cover Sheet

This document was retrieved from the Documentation and Records Management (DRM) ISEARCH System. It is intended for Information only and may not be the most recent or updated version. Contact a Document Service Center (see Hanford Info for locations) if you need additional retrieval information.

Accession #: D195065893

Document #: SD-WM-CSDD-003

Title/Desc:

SURVEILLANCE ANALYSIS COMPUTER SYSTEM SOFTWARE
DESIGN DOCUMENT

Pages: 162

ENGINEERING CHANGE NOTICE

Page 1 of 21. ECN 626411Proj.
ECN

2. ECN Category (mark one) Supplemental ☐ Direct Revision ☒ Change ECN ☐ Temporary ☐ Standby ☐ Supersedeure ☐ Cancel/Void ☐	3. Originator's Name, Organization, MSIN, and Telephone No. J. A. Glasscock, 62100, R1-01 373-4302	3a. USQ Required? ☐ Yes ☒ No	4. Date 10/02/95
	5. Project Title/No./Work Order No. "Surveillance Analysis Computer System (SACS) Software Design Description (SDD)"	6. Bldg./Sys./Fac. No. N/A	7. Approval Designator N/A
	8. Document Numbers Changed by this ECN (includes sheet no. and rev.) WHC-SD-WM-CSDD-003, Rev. 0	9. Related ECN No(s).	10. Related PO No. N/A
11a. Modification Work ☐ Yes (fill out Blk. 11b) ☒ No (NA Blks. 11b, 11c, 11d)	11b. Work Package No. N/A	11c. Modification Work Complete N/A _____ Cog. Engineer Signature & Date	11d. Restored to Original Condition (Temp. or Standby ECN only) N/A _____ Cog. Engineer Signature & Date
12. Description of Change Complete revision of document.			
13a. Justification (mark one) Criteria Change ☒ Design Improvement ☐ Environmental ☐ Facility Deactivation ☐ As-Found ☐ Facilitate Const ☐ Const. Error/Omission ☐ Design Error/Omission ☐			
13b. Justification Details To keep the SACS SDD current.			
14. Distribution (include name, MSIN, and no. of copies) SEE DISTRIBUTION SHEET		RELEASE STAMP OFFICIAL RELEASE BY WHC DATE DEC 06 1995 <i>Sta. 4</i>	

RELEASE AUTHORIZATION

Document Number: WHC-SD-WM-CSDD-003, REV 1

Document Title: Surveillance Analysis Computer System (SACS)
Software Design Document (SDD)

Release Date: 12/5/95

**This document was reviewed following the
procedures described in WHC-CM-3-4 and is:**

APPROVED FOR PUBLIC RELEASE

WHC Information Release Administration Specialist:

Kara Broz

12/5/95

TRADEMARK DISCLAIMER. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors.

This report has been reproduced from the best available copy. Available in paper copy. Printed in the United States of America. To obtain copies of this report, contact:

Westinghouse Hanford Company - Document Control Services
P.O. Box 1970, Mailstop H6-08, Richland, WA 99352
Telephone: (509) 372-2420; Fax: (509) 376-4989

SUPPORTING DOCUMENT		1. Total Pages 156
2. Title Surveillance Analysis Computer System (SACS) Software Design Document (SDD)	3. Number WHC-SD-WM-CSDD-003	4. Rev No. 1
5. Key Words Surveillance Analysis Computer System (SACS), SACS Project, Tank Farms, 101-SY, Software Design Description (SDD)	6. Author Name: J. A. Glasscock <i>J. A. Glasscock</i> Signature Organization/Charge Code 62100/R60094	
7. Abstract This document contains the Software Design Description for Phase II of the SACS Project, an Impact Level 3Q system.		
8. RELEASE STAMP <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> OFFICIAL RELEASE 3 BY WHC DATE DEC 06 1995 <i>Sta. 4</i> </div>		

[illegible]

SURVEILLANCE ANALYSIS COMPUTER SYSTEM
(SACS)

SOFTWARE DESIGN DOCUMENT
(SDD)

September 1995

BY:

J.A. Glasscock

Table of Contents

1. Introduction.....	8
1.0 Purpose.....	8
1.1 Scope.....	8
1.2 Acronyms, Abbreviations, And Definitions.....	8
1.3 Descriptions for Different Types of Design Entities.....	8
1.3.1 Visual Basic Forms.....	13
1.3.2 Database Stored Procedures.....	14
1.3.3 Data Dictionary Entries - Tables.....	14
1.3.4 Data Dictionary Entries - Columns.....	14
1.3.5 UNIX Shell Scripts.....	14
2. References.....	14
3. Functional Decomposition.....	14
3.0 Miscellaneous Design Details.....	15
3.0.1 ABOUT.FRM - frmAbout - VB Form.....	15
3.0.2 MESSAGE.FRM - frmMessages - VB Form.....	15
3.0.3 SPLASH.FRM - frmSplash - VB Form.....	15
3.1 ENTER READINGS DATA.....	15
3.1.1 Manual Data Entry.....	15
3.1.1.1 Temperature Data.....	15
3.1.1.1.1 TPFARM.FRM - frmTempSelection - VB Form.....	15
3.1.1.1.2 TPINPUT.FRM - frmTempInput - VB Form.....	15
3.1.1.1.3 TPMODIFY.FRM - frmTempModify - VB Form.....	16
3.1.1.2 Surface Level Data.....	16
3.1.1.2.1 SLINPUT.FRM - frmSurfaceLevel - VB Form.....	16
3.1.1.2.2 SLRTRV.FRM - frmSlvIget - VB Form.....	16
3.1.1.3 Interstitial Liquid Level Data.....	16
3.1.1.4 Comments.....	16
3.1.1.4.1 CMNTMOD.FRM - frmCommentInputModify - VB Form.....	16
3.1.1.4.2 OTHRCMNT.FRM - frmOtherComments - VB Form.....	17
3.1.1.4.3 READCMNT.FRM - frmReadingComment - VB Form.....	17
3.1.2 Automatic Data Entry.....	17
3.1.2.1 Temperature - TMACS.....	17
3.1.2.1.1 Process Flow.....	17
3.1.2.1.2 DoRDBMS.sh - UNIX sh File.....	18
3.1.2.1.3 ProcessRawTMACS.sh - UNIX sh File.....	18
3.1.2.1.4 start_TMACS_data_processing.sh - UNIX sh File.....	18
3.1.2.1.5 TMACS_Interface_Monitor.sh - UNIX sh File.....	19
3.1.2.1.6 TMACS_watch.sh - UNIX sh File.....	19
3.1.2.2 Surface Level Data.....	20
3.1.2.2.1 Manually Entered Readings - CASS.....	20
3.1.2.2.2 Process_new_files.sh - UNIX sh Script.....	21
3.1.2.2.3 ProcessSurfaceLevels.sh - UNIX sh Script.....	21
3.1.2.2.4 start_CASS_data_processing.sh - UNIX sh Script.....	22
3.1.2.2.5 start_KERMIT.sh - UNIX sh Script.....	22
3.1.2.2.6 Automatically Collected Readings - CASS.....	22
3.1.2.2.7 Automatically Collected Readings - TMACS.....	23

3.1.2.3 Interstitial Liquid Level - CASS.....	23
3.1.2.3.1 Process Flow	23
3.1.2.3.2 ProcessILL.sh - UNIX sh File	23
3.1.2.3.3 ProcessRAWSurfaceLevels.sh.....	23
3.2 PERFORM VALIDATION AND VERIFICATION	24
3.2.1 Automatic V&V	24
3.2.2 Manual V&V.....	24
3.2.2.1 ALLDATA.FRM - frmAllDataGS - VB Form	24
3.2.2.2 BLRETRVE.FRM - frmRetrieveBL - VB Form	24
3.2.2.3 VVGET.FRM - frmVVget - VB Form	24
3.2.2.4 VVGRAPH.FRM - frmVVGraph - VB Form	24
3.2.2.5 VVGRID.FRM - frmVVGrid - VB Form	25
3.2.2.6 VVMAIN.FRM - frmVVMMain - VB Form	25
3.2.2.7 VVSCALE.FRM - frmVVScale - VB Form	25
3.3 MAINTAIN CONFIGURATION	25
3.3.1 Physical Configuration	25
3.3.2 Baselines	26
3.3.2.1 BASELINE.FRM - frmBaseline - VB Form	26
3.3.3 Sample Frequency	26
3.3.3.1 FREQDEFN.FRM - FrmFreqDefn - VB Form	26
3.3.3.2 SAMPFREQ.FRM - FrmSamplerFreq - VB Form	26
3.3.4 Tank Group	26
3.3.4.1 CONFIG.FRM - frmConfig - VB Form	26
3.3.4.2 GROUPEDES.FRM - frmGroupDesc - VB Form	27
3.4 RETRIEVE READINGS DATA.....	27
3.4.1 Readings Retrieval Form	27
3.4.1.1 QUERY.FRM - frmQuery - VB Form	27
3.4.1.2 Text File Format.....	27
3.4.1.3 Excel Format	27
3.4.2 Report Selection Forms.....	27
3.4.2.1.1 RPTCHILD.FRM - FmReportChild - VB Form.....	27
3.4.2.1.2 RPTSEL.FRM - FmReportSelection - VB Form	27
3.4.2.1.3 SELFONT.FRM - FmSelfFont - VB Form.....	28
3.4.2.2 Frequency Violation Report Request Form	28
3.4.2.2.1 FREQ.FRM - FmFrequency - VB Form	28
3.4.2.3 Surveillance Alert Report Request Form.....	28
3.4.2.3.1 BLREPORT.FRM - frmBaselineReport - VB Form.....	28
3.4.2.4 Temperature Profile Report Request Form	28
3.4.2.4.1 TEMPRPT.FRM - frmTempReport - VB Form.....	28
3.4.2.5 Surface Level Report Request Form.....	29
3.4.2.5.1 SLVLRPT.FRM - frmSLVLRpt - VB Form.....	29
3.4.2.6 Maximum Temperature Reports Request Form	29
3.4.2.6.1 TEMP_RPT.FRM - frmMaxTempRpt - VB Form	29
3.4.2.7 Comment Report Request Form.....	29
3.4.2.7.1 CMNTRPT.FRM - frmCommentReport - VB Form.....	29
3.4.3 Printed Formats	30
3.4.3.1 Frequency Violation Report	30
3.4.3.1.1 FREQVIO.FRM - frmFrequencyViolation - VB Form	30
3.4.3.2 Surveillance Limit Report	30
3.4.3.2.1 SURALERT.FRM - frmSurveillanceAlert - VB Form	30
3.4.3.3 Temperature Profile Report	30

3.4.3.4 Temperature Comment Report	30
3.4.3.5 Surface Level Reports	30
3.4.3.6 Maximum Temperature Reports	30
3.4.3.7 Comment Report	30
3.4.4 Reports to Other Computer Systems	30
3.4.4.1 TWINS	30
3.4.4.2 TMACS	31
3.4.4.3 HDAT	31
3.5 MAINTAIN DATABASE	31
3.5.1 Database Access	31
3.5.1.1 ENTERPWD.FRM - frmEnterPWD - VB Form	31
3.5.1.2 LOGIN.FRM - frmLOGIN - VB Form	31
3.5.1.3 MDIMAIN.FRM - mdlMain - VB Form	31
3.5.1.4 PASSWORD.FRM - frmPassword - VB Form	32
3.5.2 Database Design	32
3.5.2.1 Configuration Variables	32
3.5.2.2 Data Dictionary - Tables	33
3.5.2.3 Triggers on Tables	46
3.5.2.4 Data Dictionary - Columns	48
3.6 ARCHIVE DATA	78
4. Project Reference Tables and Appendicies	79
4.0 Hardware and Software Requirements	79
4.0.1 Server	79
4.0.2 Client	79
4.1 MS-Windows Dependencies	79
4.2 Visual Basic Constants	79
4.2.1 CONSTANT.BAS File	79
4.2.2 GLOBAL.BAS File	80
4.3 Visual Basic External Products - VBX files	86
4.4 Visual Basic Forms call Visual Basic Forms	86
4.5 Visual Basic Forms call Stored Procedures	88
4.6 Stored Procedures called by Visual Basic Forms	91
4.7 Stored Procedures called by Visual Basic Forms	93
4.8 Sybase Internal Dependencies	86
4.9 Database Stored Procedures	94
4.9.1 Ip_AddMessage - Stored Procedure	94
4.9.2 Ip_AddTankToGroup - Stored Procedure	94
4.9.3 Ip_Add_MRDCNAM_NAME - Stored Procedure	95
4.9.4 Ip_Add_TMACS_rdg - Stored Procedure	95
4.9.5 Ip_DelFSSSFrow - Stored Procedure	96
4.9.6 Ip_DelFreqDefn - Stored Procedure	96
4.9.7 Ip_DeleteBaseline - Stored Procedure	97
4.9.8 Ip_DeleteCommentReport - Stored Procedure	97
4.9.9 Ip_DeleteNamedStructureComment - Stored Procedure	98
4.9.10 Ip_DeleteSensorComment - Stored Procedure	98
4.9.11 Ip_DeleteTankGroup - Stored Procedure	98
4.9.12 Ip_Delete_SLVL_RDG - Stored Procedure	99
4.9.13 Ip_Delete_TPTR_RDG - Stored Procedure	99
4.9.14 Ip_DisSurvAlertsBatchRptILL - Stored Procedure	99
4.9.15 Ip_DisSurvAlertsBatchRptSLVL - Stored Procedure	100
4.9.16 Ip_DisSurvAlertsBatchRptTPTR - Stored Procedure	101

4.9.17 lp_Drop_User_SnsrPtsRetParm - Stored Procedure	102
4.9.18 lp_GetAllFreqDefnName - Stored Procedure.....	102
4.9.19 lp_GetAllSensorPts - Stored Procedure	103
4.9.20 lp_GetAllSensorPtsCommentMaint - Stored Procedure.....	103
4.9.21 lp_GetAllSensorTypes - Stored Procedure	104
4.9.22 lp_GetAllTankGroups - Stored Procedure	105
4.9.23 lp_GetAllTankNames - Stored Procedure.....	105
4.9.24 lp_GetBaselinesILL - Stored Procedure	105
4.9.25 lp_GetBaselinesSLVL - Stored Procedure.....	106
4.9.26 lp_GetBaselinesTPTR - Stored Procedure.....	106
4.9.27 lp_GetBatchTime - Stored Procedure	107
4.9.28 lp_GetCurrentGoodSLVL - Stored Procedure	108
4.9.29 lp_GetDistMeasName - Stored Procedure.....	108
4.9.30 lp_GetFreqViaSnsrGrps - Stored Procedure.....	108
4.9.31 lp_GetFSSSTank - Stored Procedure	109
4.9.32 lp_GetFreqDefn - Stored Procedure	109
4.9.33 lp_GetFreqViaRptFit - Stored Procedure	110
4.9.34 lp_GetManualSensorTypes_SLVL - Stored Procedure	110
4.9.35 lp_GetMessages - Stored Procedure.....	110
4.9.36 lp_GetINSTRUCTRComments - Stored Procedure	111
4.9.37 lp_GetNamedStructureComment - Stored Procedure	111
4.9.38 lp_GetRDGComments - Stored Procedure.....	112
4.9.39 lp_GetSNSRComments - Stored Procedure	113
4.9.40 lp_GetSensorComment - Stored Procedure.....	113
4.9.41 lp_GetSensorGroupBySensorType - Stored Procedure	114
4.9.42 lp_GetSensorNumber - Stored Procedure	114
4.9.43 lp_GetSensorPts - Stored Procedure	114
4.9.44 lp_GetSensorTypes - Stored Procedure	115
4.9.45 lp_GetSensorTypes_GST - Stored Procedure	116
4.9.46 lp_GetSensorTypes_SLVL - Stored Procedure	116
4.9.47 lp_GetSnsrPtsILL - Stored Procedure	117
4.9.48 lp_GetSnsrPtsSLVL - Stored Procedure	117
4.9.49 lp_GetSnsrPtsTPTR - Stored Procedure.....	118
4.9.50 lp_GetSnsrRdgGroup - Stored Procedure	118
4.9.51 lp_GetSnsrRdgGroupByTank - Stored Procedure	119
4.9.52 lp_GetSnsrSpacing - Stored Procedure	119
4.9.53 lp_GetSurvAlertsBatchRpt - Stored Procedure.....	120
4.9.54 lp_GetSurvAlertsRptILL - Stored Procedure	120
4.9.55 lp_GetSurvAlertsRptSLVL - Stored Procedure.....	121
4.9.56 lp_GetSurvAlertsRptTPTR - Stored Procedure.....	122
4.9.57 lp_GetTankAndSensorComment - Stored Procedure	123
4.9.58 lp_GetTankAndSensorCommentCnt - Stored Procedure.....	124
4.9.59 lp_GetTankGroupDesc - Stored Procedure.....	124
4.9.60 lp_GetTanksInGroup - Stored Procedure.....	124
4.9.61 lp_GetTankName - Stored Procedure	125
4.9.62 lp_GetTimeMeasCode - Stored Procedure	125
4.9.63 lp_GetTptrMeasName - Stored Procedure	126
4.9.64 lp_GetVersion - Stored Procedure	126
4.9.65 lp_Get_COLCTN_NO - Stored Procedure.....	126
4.9.66 lp_get_farm - Stored Procedure.....	127
4.9.67 lp_Get_MRDCNAM_NAME - Stored Procedure	127

4.9.68	lp_Get_MRDCNAM_NAME_byTank - Stored Procedure	128
4.9.69	lp_Get_Num_TPTR_Risers - Stored Procedure	128
4.9.70	lp_Get_Num_TPTR_Risers_byTank - Stored Procedure	129
4.9.71	lp_Get_Num_TPTR_Sensors - Stored Procedure	129
4.9.72	lp_Get_RDG_DEVICE_NAME - Stored Procedure	130
4.9.73	lp_Get_TPTR_Data - Stored Procedure	130
4.9.74	lp_Get_TPTR_RDGS - Stored Procedure	130
4.9.75	lp_Get_TPTR_Risers - Stored Procedure	131
4.9.76	lp_Get_TPTR_Risers_byTank - Stored Procedure	131
4.9.77	lp_Get_TPTR_Sensors - Stored Procedure	132
4.9.78	lp_Get_TPTR_Sensors_byTank - Stored Procedure	132
4.9.79	lp_Get_TPTR_Time - Stored Procedure	133
4.9.80	lp_Get_User_SnsrPtsRetParm - Stored Procedure	133
4.9.81	lp_InsertBaseline - Stored Procedure	134
4.9.82	lp_InsertFSSSFRow - Stored Procedure	135
4.9.83	lp_InsertFreqDefn - Stored Procedure	135
4.9.84	lp_InsertNamedStructureComment - Stored Procedure	136
4.9.85	lp_InsertSLVL - Stored Procedure	136
4.9.86	lp_InsertSensorComment - Stored Procedure	137
4.9.87	lp_InsertTPTR - Stored Procedure	137
4.9.88	lp_Is_MRDCNAM_NAME_Valid - Stored Procedure	138
4.9.89	lp_LastSLVLReading - Stored Procedure	138
4.9.90	lp_Load_User_SnsrPtsRetParm - Stored Procedure	138
4.9.91	lp_LockRpt - Stored Procedure	139
4.9.92	lp_MakeFreqVioRpt - Stored Procedure	140
4.9.93	lp_MakeTPTRCommentReport - Stored Procedure	140
4.9.94	lp_ModifyNamedStructureComment - Stored Procedure	141
4.9.95	lp_ModifySensorComment - Stored Procedure	141
4.9.96	lp_RemoveTankFromGroup - Stored Procedure	142
4.9.97	lp_RetrieveData - Stored Procedure	142
4.9.98	lp_RetrieveILL - Stored Procedure	143
4.9.99	lp_RetrieveILLDataBatch - Stored Procedure	144
4.9.100	lp_RetrieveSLVL - Stored Procedure	145
4.9.101	lp_RetrieveSLVLData - Stored Procedure	146
4.9.102	lp_RetrieveSLVLDataBatch - Stored Procedure	146
4.9.103	lp_RetrieveTPTR - Stored Procedure	147
4.9.104	lp_RetrieveTPTRDataBatch - Stored Procedure	148
4.9.105	lp_RetrieveTPTR_byTC - Stored Procedure	149
4.9.106	lp_RetrieveVVBaselines - Stored Procedure	150
4.9.107	lp_TemperatureReport - Stored Procedure	150
4.9.108	lp_UnlockRpt - Stored Procedure	151
4.9.109	lp_UpdateBaseline - Stored Procedure	151
4.9.110	lp_UpdateSLVL - Stored Procedure	152
4.9.111	lp_UpdateTPTR - Stored Procedure	153
4.9.112	lp_Updt_SuspDataStatCode - Stored Procedure	153

List of Tables

Table 1. Acronyms and Abbreviations	8
Table 2. Definitions	10

Table 3. Sybase Database Configuration Variables	32
Table 4. Triggers on Tables	46
Table 5. Visual Basic forms that call other Forms	86
Table 6. Visual Basic forms called by other Forms	87
Table 7. Visual Basic forms call Stored Procedures	88
Table 8. Stored Procedures called by Visual Basic Forms	91
Table 9. Stored Procedures called by UNIX Scripts	94

1. Introduction

1.0 Purpose

This Software Design Description (SDD) is the primary document describing the design of the Surveillance Analysis Computer System (SACS). This system satisfies the requirements identified in the SACS Software Requirements Specification, WHC-SD-WM-CSRS-007. The SACS is an Impact Level 3Q system.

1.1 Scope

This SDD describes the design of the SACS and follows the *Standard Engineering Practices (WHC-CM-6-1)*, *Software Practices (WHC-CM-3-10)* and *Quality Assurance (WHC-CM-4-2, QR 19.0)* policies. This SDD includes sections on definitions, references, and descriptions of the design entities within SACS. The design entities are organized according to the functional breakdown of the SRS. There are cross-reference tables in Section 4.

This SDD is current through SACS revision 3.5.1, September, 1995.

The SACS defined in this document stores information in a database on tank temperatures, surface levels, and interstitial liquid levels. The data are received from several sources. This information is retrieved by the customer through a PC-based interface and is then available to a number of other software tools.

1.2 Acronyms, Abbreviations, And Definitions

Table 1. Acronyms and Abbreviations

BCSR	Boeing Computer Services Richland
CASS	Computer Automated Surveillance System
DOE	United States Department of Energy
ERD	Entity-relationship Diagram
FIC	Food Instrument Corporation
H DAT	Hand-Held Data Acquisition for Tank Farms
HLAN	Hanford Local Area Network
ILL	Interstitial Liquid Level
ISS	Information and Scientific Systems
LOW	Liquid Observation Well
OSD	Operating Specification Document
OSR	Operational Safety Requirements

QI	Quality Indicator
SACS	Surveillance Analysis Computer System
SLVL	Surface Level
SQL	Structured Query Language
TC	Thermocouple
TCP/IP	Transmission Control Protocol / Internet Protocol
TMACS	Tank Monitor and Control System
TPTR	Temperature
TWINS	Tank Waste Information Network

Table 2. Definitions

Computer Automated Surveillance System	The Computer Automated Surveillance System (CASS) is a computer system that collects tank farm surveillance data from the tank farms. CASS often refers to the entire surveillance system, which includes the central facility, substation microcomputers, drywell vans, and field sensors.
Configuration Data	Configuration data are data that define the characteristics and state of an entity. Configuration data are not measured values. Examples of configuration data are tank characteristics, baseline values, and criteria limits.
Criteria/ criterion	Criteria are the monitoring characteristics against which readings are compared. See Surveillance Criteria.
Data Collection System	A data collection system is the hardware, software, and procedures related to gathering, processing, storing, reporting, and archiving a specific type of data.
Date_Time	A date and a time treated as a single data element.
Department of Energy	The Department of Energy (DOE) is the United States agency responsible for the oversight of operations at the Hanford Site.
Double-Shell Tank	A double-shell tank is an underground storage tank constructed of reinforced concrete with a primary steel liner, an air space, and a secondary steel liner. These tanks have one million gallon capacity and are used to store high level radioactive wastes.
Drywell	A drywell is a vertical bore hole located outside a waste storage tank or crib. A drywell has a steel casing generally six inches in diameter and is drilled to various depths. It is called a drywell because it does not reach the water table. A drywell is drilled to allow monitoring for detection of leaks and movement of existing radionuclides in the soil.
Entity	An entity is a person, place, thing, or concept that has characteristics of interest to the enterprise.
Entity-Relationship Diagram	An entity-relationship diagram (ERD) is a figure, following a certain style, that displays business entities and their relationships with each other. ERDs are used to explain the business model of a (proposed) computer system.
File	A file is a collection of records in a single place.
File Transfer Protocol	File Transfer Protocol (FTP) is a TCP/IP protocol for transferring files from one computer to another.
Graph	A graph is a plot or graphic display suitable for analyst interpretation. A graph may appear on-screen, on paper, or in a computer file.
Hand-Held Data Acquisition for Tank Farms	Hand-Held Data Acquisition for Tank Farms (HDAT) is a system that improves manual data collection by utilizing hand-held computers and bar code technology.

Hanford Local Area Network	The Hanford Local Area Network (HLAN) is a local area network on the Hanford Site.
Information and Scientific Systems	Information and Scientific Systems (ISS) is an organization of BCSR that provides support and personnel for development and maintenance of information and scientific computer systems.
Limits	Values used as operating limits set forth by the engineers. They are also known as "Bounding Values" when displaying, reporting or analyzing the data for a given tank. The "Limits" or "Bounds" are used when establishing alert criteria.
Liquid Level	The liquid level is the height in inches from the reference bottom of a waste storage tank to the top of the supernatant liquid. Liquid levels are monitored in tanks to calculate waste volumes and to identify intrusions and leakage.
Liquid Observation Well	LOW. A thin tube, closed on the bottom, that is inserted into a waste storage tank through a riser. Radiation sampling equipment is inserted in this tube to sample conditions inside the tank.
Operating Specification Document	An Operating Specification Document (OSD) is a document issued by WHC to define criteria related to various activities performed by Tank Farm Operations.
Operational Safety Requirement	Operational Safety Requirements (OSRs) are mandatory criteria related to safety concerns. OSRs define operational, surveillance and design controls to ensure safe operation. They are established in Safety Analysis Reports (SARs).
Probe	A probe is a mobile carrier of one or more sensors. It may refer to the sensor itself, if mobile. Examples of probes are gamma probes and temperature probes used in LOWs.
Remote Access	Remote access is access via HLAN to a computer system.
Report	A report is a formatted output suitable for analyst interpretation. A report may appear on-screen, on paper, or in a computer file.
Riser	A riser is a pipe emerging through the top of an underground waste tank or from a transfer line. Risers are identified by number on engineering drawings and in <i>Double-shell Underground Waste Storage Tanks - Riser Survey, WHC-SD-RE-TI-093</i> , and in <i>Riser Configuration Document for Single Shell Tanks, WHC-SD-RE-TI-053</i> . Risers may be used for various monitoring and operation purposes, such as temperature measurements or LOW insertion.
Sensor	A sensor is a instrument that converts physical states or conditions into a signal representative of the state or condition.
Single-Shell Tank	A single-shell tank is an underground waste storage tank of reinforced concrete with a steel liner on the sides and bottom.
String	A series of characters treated as a group.
Structured Query Language	Structured Query Language (SQL) is a standardized language for requesting data from a database.

Surface-Level	Surface-level is the measurement of the top of the waste measured from the bottom of the tank in inches. These measurements are taken to allow calculation of waste volumes and identify intrusions or leaks.
Surveillance	Surveillance is the act of monitoring or observing to verify whether an item or activity conforms to specified requirements.
Surveillance Analysis Computer System	Surveillance Analysis Computer System (SACS) is a computer system providing electronic storage and report generation of tank farm surveillance data. The SACS includes the software, the database, and the documentation.
Surveillance Criteria	Surveillance criteria are monitoring criteria against which readings are compared.
Suspect Data	Suspect data are data that a SDA technical analyst has reviewed for correctness, but has chosen to suspend final judgment.
Sybase	Sybase is a commercial relational database management software package. It includes software for data entry, database queries, and reports from the database. A registered trademark of Sybase, Inc.
Tank Farm	A tank farm is a set of waste storage tanks. A tank farm is identified by a 1 or 2 character name, e.g., A, AX, or AP.
Tank Farm Surveillance Data	Tank farm surveillance data are data gathered to verify integrity and safe operation of waste storage tanks.
Tank Integrity	Tank integrity is a condition associated with a waste storage tank. Tank integrity may be sound or assumed leaker.
Tank Monitor and Control System	Tank Monitor and Control System (TMACS) monitors tank farm instrumentation in 200 area waste tanks, displays current values and trends of these instruments, and provides alarm annunciation for the Tank Farm Operators. The TMACS computers and operator terminals are located in the Computer Automated Surveillance System (CASS) Central Facility.
Tank Temperature	Tank temperature refers to the temperature sensor reading obtained at a specific location associated with a waste storage tank or related equipment.
Tank Temperature Data	Tank temperature data are tank temperature measurements. These readings are obtained manually and electronically from thermocouples and other temperature measurement devices associated with the waste storage tanks.
Temperature Probe	A temperature probe is a structure containing an assembly of temperature sensors. The assembly is composed of one or more temperature sensors and the wiring that carries the signals generated by the sensors. Also called a Thermocouple Tree.
Thermocouple	A thermocouple is an electrical device used to measure temperatures.

Transmission Control Protocol / Internet Protocol	Transmission Control Protocol / Internet Protocol (TCP/IP) is the communication protocol used to connect the Sun workstations to the network.
Trigger	Specific to relational databases: a type of stored procedure that is executed automatically when a user attempts a specified modification statement on a specified table.
Tank Waste Information Network	TWINS is run by Battelle-NW for the US DOE. It retrieves information from several waste-storage databases and shows that information on a common screen.
Valid Data	Valid data are values within the range possible for a sensor.
Washington State Department of Ecology	The Washington State Department of Ecology is the state agency responsible for assuring that Washington State regulations are adhered to by the Hanford Site.
Waste Storage Tank	See Double-Shell Tank and Single-Shell Tank.
Westinghouse Hanford Company	Westinghouse Hanford Company (WHC) is the current site operations and engineering contractor for the Hanford Site.

1.3 Descriptions for Different Types of Design Entities

IEEE Standard 1087-1086 requires definition of 10 attributes for each design entity. These attributes have been applied differently to the different kinds of design entities used in the SACS system.

- Two attributes are in the title of the section that describes the module: Identification and Type.

The following are defined in the text of the design entity.

- Purpose
- Function
- Subordinates
- Dependencies
- Interface
- Resources
- Processing
- Data

1.3.1 Visual Basic Forms

The visual display of the VB Forms is not given in this document.

- Purpose - The purpose of the form is explained in this section.
- Function
- Subordinates - This information is kept in a cross-reference table in Section 4.
- Dependencies
- Interface - Not used for VB Forms.
- Resources - External programs that are used by this form are listed here. Examples are VBX files that are used.
- Processing - Special processing by the form is explained here.
- Data - Special data arrays are described here. Especially the V&V forms with its passed arrays.

1.3.2 Database Stored Procedures

- Purpose - The purpose of the stored procedure is explained here.
- Function - The function of the stored procedure is explained here. Often the Purpose and the Function are the same.
- Subordinates - Stored Procedures called by this one are listed.
- Dependencies - Database tables referenced by this procedure are listed here.
- Interface - Data received and returned are listed here.
- Resources
- Processing

1.3.3 Data Dictionary Entries - Tables

Each table in the database has an entry in the Data Dictionary. The structure is:

- The name of the table.
- Description: A description of why the table exists.
- Columns: The columns in the table.

1.3.4 Data Dictionary Entries - Columns

Each column in each table has an entry in the data dictionary. The structure is like:

- The name of the column. E.g. `FREQ_NO`.
- Description: A description of the data element stored in this column.
- Used in: A list of the tables where this data element is used.
- Alias (optional): Other places where the same data element has a different name.
- Type: The type of data element. Examples are `smallint`, `int`, `char 30`.
- Default: The default definition used for this column within the database. If there is no default, a NULL is printed.
- Rule: The rule (or domain) definition used for this column within the database. If there is no rule, a NULL is printed.
- Values (optional): Where the number of possible values is small, a list is given along with the meaning for each value.
- Source: The source of this data element. Code within `lp_InsertFreqDefn`.

1.3.5 UNIX Shell Scripts (UNIX is a registered trademark of AT&T)

- Purpose - The purpose of the shell script is explained here.
- Function - The function of the shell script is explained here.
- Subordinates
- Dependencies
- Interface
- Resources
- Processing
- Data

2. References

SACS Software Requirements Specification (WHC-SD-WM-CSRS-007) defines the requirements as of Software installation 03.05.01, September, 1995.

SACS Entity Relationship Diagram (ERD) Document (WHC-SD-WM-CSWD-037) gives a look at the business system as of 1992.

3. Functional Decomposition

The different functional area and requirements are given in the SRS. The Visual Basic forms that relate to each functional area are listed.

3.0 Miscellaneous Design Details

3.0.1 ABOUT.FRM - frmAbout - VB Form

- Purpose

Display the current version and compile date for the Visual Basic forms.

- Function

Displays the current version and compile date for the SACS.exe. Retrieves system information about processors, modes, and memory and displays that.

- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.0.2 MESSAGE.FRM - frmMessages - VB Form

- Purpose

Used to enter the text of the message of the day.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.0.3 SPLASH.FRM - frmSplash - VB Form

- Purpose

The introductory form that the customer looks at while waiting for the program to start.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1 ENTER READINGS DATA

3.1.1 Manual Data Entry

3.1.1.1 Temperature Data

3.1.1.1.1 TPFARM.FRM - frmTempSelection - VB Form

- Purpose

Allows the user to select whether to enter or modify temperature data; and to specify the farm, date, and time for the data.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.1.2 TPINPUT.FRM - frmTempInput - VB Form

- Purpose
Input temperature readings and related information.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data - Reading values, type of sensor, type of reading device.

3.1.1.1.3 TPMODIFY.FRM - frmTempModify - VB Form

- Purpose
Allows user to modify temperature readings and comments.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.2 Surface Level Data

3.1.1.2.1 SLINPUT.FRM - frmSurfaceLevel - VB Form

- Purpose
Input and modify surface-level readings.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.2.2 SLRTRY.FRM - frmSlvlget - VB Form

- Purpose
Used to set parameters for selecting the surface levels to modify. Similar in appearance to the query form.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.3 Interstitial Liquid Level Data

Currently there are no forms used for manual entry of ILL data.

3.1.1.4 Comments

3.1.1.4.1 CMNTMOD.FRM - frmCommentInputModify - VB Form

- Purpose

Input and modify comments for tanks and sensors.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.4.2 OTHRCMNT.FRM - frmOtherComments - VB Form

- Purpose

Display the comments for tanks and sensors.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.1.4.3 READCMNT.FRM - frmReadingComment - VB Form

- Purpose

Allows the user to enter a comment for a single reading.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.1.2 Automatic Data Entry

The basic design for automatic data entry into SACS is:

An external computer sends a file to SACS.

A watchdog process on the SACS computer:

- detects the presence of the file
- edits the file into a form that the database expects
- inserts the data into the database
- moves the processed file to a storage directory.

The specifics for each interface are described in the following sections.

3.1.2.1 Temperature - TMACS

3.1.2.1.1 Process Flow

Data transfers from TMACS occur around midnight each evening. The current `dst_data_yyyy_mmdd_hhmm.rdbms` and `sst_data_yyyy_mmdd_hhmm.rdbms` files made by TMACS are moved to a directory where the SACS watchdog process can see them (`/home/TMACS/G2` on `tfs8`). The watchdog process is started by the `start_TMACS_data_processing.sh` script. When a file is detected the `DoRDBMS.sh` script runs. The files are edited for insertion into the SQL database. The files are moved to the `DONE` subdirectory after processing.

The TMACS_UNTAG table contains the tag names of sensors that TMACS reports but SACS does not want. (The customer has either decided not to store these data in SACS or SACS is not configured to store these data.) Ordinarily when a reading is not inserted into SACS an error line is created; readings that do not get stored because they are in the TMACS_UNTAG table do not create error messages.

3.1.2.1.2 DoRDBMS.sh - UNIX sh File

~ Purpose

Processes the TMACS *.RDBMS files into the SACS database.

~ Function

Load the *.RDBMS files into the SACS database when called by the watchdog process.

~ Subordinates

None.

~ Dependencies

None.

~ Interface

None.

~ Resources

ISQL.

~ Processing

Edits the data file so that it can be inserted into the SACS database.

Load the files into the database using ISQL.

Sends a mail message of the problems to the system administrator.

~ Data

TMACS *.RDBMS files.

3.1.2.1.3 ProcessRawTMACS.sh - UNIX sh File

~ Purpose

Processes the TMACS continuous* files into the SACS database.

~ Function

Load the continuous* files into the SACS database when called by the watchdog process.

~ Subordinates

None.

~ Dependencies

None.

~ Interface

None.

~ Resources

ISQL.

~ Processing

Edits the TMACS file so that it can be inserted into the SACS database.

Runs the script to load the data into the database.

Sends a mail message with any errors to the system administrator.

~ Data

TMACS continuous* files.

3.1.2.1.4 start_TMACS_data_processing.sh - UNIX sh File

~ Purpose

Starts the other scripts that process data files from TMACS.

~ Function

Start the appropriate script when files are received from TMACS.

- Subordinates
DoRDBMS.sh
ProcessRawTMACS.sh
- Dependencies
None.
- Interface
None.
- Resources
None.
- Processing
Sleeps 300 seconds.
Looks for data files. If any are present, calls the appropriate processing script.
Moves the processed files to a subdirectory.
- Data
None.

3.1.2.1.5 TMACS_Interface_Monitor.sh - UNIX sh File

- Purpose
Used to monitor the file transfers between TMACS and SACS.
- Function
Looks for discrepancies in sizes between the files on SACS and the files on TMACS.
- Subordinates
None.
- Dependencies
None.
- Interface
None.
- Resources
None.
- Processing
Compares the size of files in the SACS and TMACS file systems. Files with the same name should have the same size. Reports any differences.
- Data
sst_data* and dst_data* files are compared.

3.1.2.1.6 TMACS_watch.sh - UNIX sh File

- Purpose
Used to monitor the file transfers between TMACS and SACS.
- Function
Looks for discrepancies in sizes between the files on SACS and the files on TMACS.
- Subordinates
None.
- Dependencies
None.
- Interface
None.
- Resources
mount mfs command.
- Processing
Mounts the TMACS disks.
Lists the *.RDBMS files in the CURRENT, HISTORY, and SACS directories.
Mails the results to the system administrator.

- o Data
- *.RDBMS files.

3.1.2.2 Surface Level Data

3.1.2.2.1 Manually Entered Readings - CASS

3.1.2.2.1.1 Process Flow

Transfers of manual surface level are initiated upon completion of operator input of the manual surface levels. When the reading entry is accepted by the operator, a file is written and a request is sent to the SACS QMGR process (on CASS) to initiate the data transfer. The name of the file being transferred is of the form SL_mmddhhmmss where mm=month, dd=day, hh=hour, etc. The format of the file is the same as SUN_FILE.

The Kermit client process (xkermi) is initiated by the SACS QMGR process on CASS whenever there is a data file to transfer. The xkermi CLI macro receives a fully qualified file name and runs a C program (findex) to determine the type of file to be transferred. This information is used by the xkermi CLI macro to write a "take" file which specifies what directory (ILL or SurfaceLevels) the data are to be transferred to on SACS. The "take" file also specifies that a duplicate of the original data is also transferred; the duplicate has the same name as the original with "CHECK" appended to the name.

The communications link is provided by a direct serial connection from the SACS computer (tfs8) to the T-bar and thence to each of the Data General Systems (A & B) on con17. The Kermit protocol is supported over this link with 9600 baud, even parity, and level 3 error correction.

The Kermit Server process runs on the SACS computer and is used to receive incoming Kermit file transfers. The process is run as a system daemon, owned by root. The process is started at boot time and as the Kermit client on the CASS system never sends a FIN, the process stays up ready to receive data.

The start_CASS_data_processing.sh script is started when the computer boots up. This process scans for new files every 5 minutes and checks on the operation of Kermit every 5 minutes. If Kermit is not running it tries to re-start it.

The process_new_files process checks for arrival of data files and the corresponding check files. If a matching pair of data and check files are found, the data file is tar'd to archive the information. Finally, the data is processed with the appropriate script.

The operation of the process is controlled with the DataDirectories file which specifies the directories to look in for data arrival as well as the processing script and tar location. Taking surface level information as an example, Surface Level Data files will be looked for in the /home/CASS/DATA/SurfaceLevels directory; Surface Level information will be processed with the /home/CASS/PROCESS/ProcessSurfaceLevels.sh script and the raw data will be appended to the tar file /home/CASS/TAR/SurfaceLevels.

Mismatches occur when there is no data file, no check file, or when the files do not match. In any of these cases, the files which were received are transferred to the /home/CASS/ATTENTION directory and mail is sent to the individuals responsible for the process.

The ProcessSurfaceLevels process takes individual data files transferred from CASS and processes them into SACS. For each new data file to be processed, a TSQL instance file is created and then executed. This instance file performs all work necessary to temporarily hold, edit, select, and insert the surface level information into SACS. Both automatic and manual readings are treated in the same manner.

3.1.2.2.2 Process_new_files.sh - UNIX sh Script

- Purpose

Checks for and starts processing of new CASS files.

- Function

Checks that CASS files have been received with check files; stores files in appropriate directory.

- Subordinates

None.

- Dependencies

None.

- Interface

None.

- Resources

UNIX sh script.

- Processing

Checks for new files in a directory pointed to by the DataDirectories file.

Checks that the checkfile has been received.

Archives the file in a subdirectory.

Sends mail to the system administrator if there is a problem.

- Data

DataDirectories file contains a list of the directories to look for data.

3.1.2.2.3 ProcessSurfaceLevels.sh - UNIX sh Script

- Purpose

Process all CASS surface level data into SACS and RAW_DATA.

- Function

Store in SACS a daily reading from the CASS hourly readings.

Store all the CASS readings into RAW_DATA.

- Subordinates

ProcessRawSurfaceLevels.sh

- Dependencies

None.

- Interface

None.

- Resources

None.

- Processing

Edit the received file into something that can be stored in SACS. Many edits are performed to remove '***' and fix tank names, etc.

Load the data into a temporary table #SURFACE_LEVEL_TEMP, and choose the daily reading.

Start the Surveillance Limits Report.

Start the Frequency Violation Report.

Start the script to load RAW_DATA.

- Data

SUN_FILES from CASS.

3.1.2.2.4 start_CASS_data_processing.sh - UNIX sh Script

- Purpose

Periodically start the files that actually process CASS data.

- Function

Every 300 seconds wake up; start process_new_files.sh; check the Kermit is OK (if not, re-start it).

- Subordinates

process_new_files.sh

- Dependencies

UNIX sh script.

- Interface

None.

- Resources

None.

- Processing

See Function.

- Data

None.

3.1.2.2.5 start_KERMIT.sh - UNIX sh Script

- Purpose

Starts the Kermit process.

- Function

Keep the Kermit process alive to receive CASS data.

- Subordinates

None.

- Dependencies

None.

- Interface

None.

- Resources

UNIX sh script.

- Processing

Do Forever

If there is no indication the operations should be suspended, then
start up kermit in server mode to receive files sent from CASS

If a core file exists in the ~/Data directory, remove it.

End If

End do

- Data

None.

3.1.2.2.6 Automatically Collected Readings - CASS

3.1.2.2.6.1 Process Flow

A scheduled CLI macro on CASS extracts automatic surface level information from the CASS 72 hour data repository and sends a request to the SACS QMGR process (on CASS) to initiate the data transfer. The data are transferred in a file named SUN_FILE.

See §3.1.2.2.1.1 Process Flow for the processes used.

3.1.2.2.7 Automatically Collected Readings - TMACS

See §3.1.2.1.1 Process Flow for the processes used to handle surface-level readings. These processes are the same as the temperature readings.

3.1.2.3 Interstitial Liquid Level - CASS

3.1.2.3.1 Process Flow

Transfers of ILL data are initiated upon review of the information by a surveillance engineer. When the information is reviewed, a file is written and a request is sent to the SACS QMGR process (on CASS) to initiate the data transfer. The name of the file being transferred is of the form TALL_ILL_mmdd_hhmmss where mm=month, dd=day, hh=hour, etc.

See §3.1.2.2.1.1 Process Flow for the general transfer processes used.

The ProcessILL.sh process takes individual data files transferred from CASS and processes them into SACS. For each new data file to be processed, a TSQL instance file is created and then executed. This instance file performs all work necessary to temporarily hold, edit, select, and insert the ILL information into SACS.

3.1.2.3.2 ProcessILL.sh - UNIX sh File

- Purpose

Store ILL readings in the SACS database.

- Function

Edit the file received from CASS and store it in the SACS ILL_VALUE table.

- Subordinates

None.

- Dependencies

None.

- Interface

None.

- Resources

ISQL.

- Processing

Edit the file received from CASS so it can be run into SACS.

Load the file into SACS.

Move the processed file to the DONE directory.

- Data

CASS ILL data files.

3.1.2.3.3 ProcessRAWSurfaceLevels.sh

- Purpose

- Function

- Subordinates

- Dependencies

- Interface

- Resources

- Processing

- Data

3.2 PERFORM VALIDATION AND VERIFICATION

3.2.1 Automatic V&V

This function is handled in two places: triggers on the SLVL_READING and TPTR_READING tables; and in the TMACS Interface scripts.

The table insert triggers will set the SUSP_DATA_STAT_CODE to 'S' if the values is near zero, typically reading_value ≤ 1 . Yes, negative values are suspicious. The modify triggers have no such restriction.

The TMACS Interface will set the SUSP_DATA_STAT_CODE to 'S' if the SAMPLER_STATUS is 'UNKNOWN'. This interface relies on the trigger code to set the SUSP_DATA_STAT_CODE value for readings near zero.

3.2.2 Manual V&V

3.2.2.1 ALLDATA.FRM - frmAllDataGS - VB Form

- Purpose

Used to make all of the readings in the VVGrid as good or suspect.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.2.2.2 BLRETRVE.FRM - frmRetrieveBL - VB Form

- Purpose

Used to select a specific baseline to display on the V&V form.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.2.2.3 VVGET.FRM - frmVVget - VB Form

- Purpose

Allows the user to specify the parameters for the readings to retrieve for V&V activity.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.2.2.4 VVGRAPH.FRM - frmVVGraph - VB Form

- Purpose

Holds the graph for the V&V activity. The ChartBuilder product actually does the display.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.2.2.5 VVGRID.FRM - frmVVGrid - VB Form

- ~ Purpose

Holds the grid for the V&V activity.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.2.2.6 VVMAIN.FRM - frmVVMMain - VB Form

- ~ Purpose

Allows the user to set the parameters for data that will be displayed in the graph and grid for the V&V activity.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.2.2.7 VVSCALE.FRM - frmVVScale - VB Form

- ~ Purpose

Allows the user to set date and value limits on the display window for V&V data.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.3 MAINTAIN CONFIGURATION

3.3.1 Physical Configuration

The customer does not maintain the physical configuration as of Revision 03.05.01. The customer provides the information needed for the SACS developers to enter the physical configuration. The SACS developers use a variety of SQL and UNIX scripts to enter the information. No readings can be stored in the database unless the physical configuration to support that sensor has been stored in the database.

3.3.2 Baselines

3.3.2.1 BASELINE.FRM - frmBaseline - VB Form

- Purpose

Used to enter, modify, and delete the baseline values for temperature and surface level readings.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.3.3 Sample Frequency

3.3.3.1 FREQDEFN.FRM - frmfreqDefn - VB Form

- Purpose

Used to define and maintain the definition for a sample frequency. It does not apply this frequency to any locations.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.3.3.2 SAMPFREQ.FRM - frmSamplerFreq - VB Form

- Purpose

Used to create the relationship between a sample space, a type of sampler, and a frequency.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.3.4 Tank Group

3.3.4.1 CONFIG.FRM - frmConfig - VB Form

- Purpose

Used to add and remove tanks from reporting groups.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.3.4.2 GROUPDES.FRM - frmGroupDesc - VB Form

- Purpose
Used to add, maintain, and remove reporting groups.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4 RETRIEVE READINGS DATA

3.4.1 Readings Retrieval Form

3.4.1.1 QUERY.FRM - frmQuery - VB Form

- Purpose
The main data retrieval form for the user. Used to select groups of data to report.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.1.2 Text File Format

The description of the text file format is in the SRS. The code to store data in this format is in the QUERY.FRM form.

3.4.1.3 Excel Format

The description of the text file format is in the SRS. The code to store data in this format is in the QUERY.FRM form.

3.4.2 Report Selection Forms

3.4.2.1.1 RPTCHILD.FRM - FmReportChild - VB Form

- Purpose
Blank form used in reporting??.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.2.1.2 RPTSEL.FRM - FmReportSelection - VB Form

- Purpose
Contains a menu of pre-coded reports that the user can select.
- Function
Contains buttons that will call up additional forms that allow the user to specify the data to be reported.

- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.4.2.1.3 SELFONT.FRM - FrmSelFont - VB Form

- Purpose

Allows the user to choose fonts for printing VB-generated reports. Not used with Crystal Reports.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.4.2.2 Frequency Violation Report Request Form

3.4.2.2.1 FREQ.FRM - FrmFrequency - VB Form

- ~ Purpose

Used to retrieve the Frequency Report from a batch job; run the Frequency Report for special queries.

- ~ Function
- ~ Subordinates
- ~ Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.4.2.3 Surveillance Alert Report Request Form

3.4.2.3.1 BLREPORT.FRM - frmBaselineReport - VB Form

- Purpose

Print the Surveillance Alert Report that has already been created; start the Surveillance Alert Report for special queries.

- ~ Function
- ~ Subordinates
- Dependencies
- ~ Resources
- ~ Processing
- ~ Data

3.4.2.4 Temperature Profile Report Request Form

3.4.2.4.1 TEMPRPT.FRM - frmTempReport - VB Form

- Purpose

Allows the user to enter parameters for the Temperature profile report. The results of the query are then printed.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.2.5 Surface Level Report Request Form

3.4.2.5.1 SLVLRPT.FRM - frmSLVLRpt - VB Form

- Purpose

Allows the user to specify the tank and other parameters for the surface level report. Some of these parameters are used by Crystal Reports to produce the report.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.2.6 Maximum Temperature Reports Request Form

3.4.2.6.1 TEMP_RPT.FRM - frmMaxTempRpt - VB Form

- Purpose

Allows the user to enter the parameters for the Maximum Temperature Report. The parameters are used to run a stored procedure; the results of this are sent to Crystal Reports.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.2.7 Comment Report Request Form

3.4.2.7.1 CMNTRPT.FRM - frmCommentReport - VB Form

- Purpose

Used to create a comment report and direct the output to different locations in different formats.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.3 Printed Formats

3.4.3.1 Frequency Violation Report

3.4.3.1.1 FREQVIO.FRM - frmFrequencyViolation - VB Form

- Purpose

Used to display the Frequency Violation Report on screen. Contains a control to let the customer print the report displayed in this window.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.3.2 Surveillance Limit Report

3.4.3.2.1 SURALERT.FRM - frmSurveillanceAlert - VB Form

- Purpose

Used to display the Surveillance Alert report on the screen. The user may print this report after looking at the screen copy.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.4.3.3 Temperature Profile Report

This report is displayed using VB code.

3.4.3.4 Temperature Comment Report

This report is displayed using Crystal Reports.

3.4.3.5 Surface Level Reports

These reports are displayed using Crystal Reports.

3.4.3.6 Maximum Temperature Reports

These reports are displayed using Crystal Reports.

3.4.3.7 Comment Report

These reports are displayed using Crystal Reports.

3.4.4 Reports to Other Computer Systems

3.4.4.1 TWINS

TWINS uses 4 views in the SACS database to support their data model and assist them in retrieving data. These views are written and maintained by the TWINS team - the SACS team installs them and checks that they won't crash the system. These views are:

- svl_sgl_measurement_v - for a single surface level measurement device in a tank.

- Temp_col_measurement_v - for a collection of temperature measurement devices in a tank, e.g. a tree.
- Temp_sgl_measurement_v - for a single temperature measurement device in a tank, e.g. embedded sensors.
- Ill_sgl_measurement_v - for a single ILL measurement device in a tank.

3.4.4.2 TMACS

TMACS uses an SQL-bridge product to retrieve data from SACS. This product talks directly to the database and allows TMACS to retrieve the latest surface level readings through the Ip_LastSLVLRReading stored procedure. This product expects column headers from ISQL that exactly match (including case) the column headers in the bridge code; and ASCII data. That is why the stored procedure converts everything to ASCII and makes certain column names before it returns the surface level reading.

3.4.4.3 HDAT

None designed at this time.

3.5 MAINTAIN DATABASE

3.5.1 Database Access

3.5.1.1 ENTERPWD.FRM - frmEnterPWD - VB Form

- Purpose

Used to re-enter the user password after the connection to the database has timed out.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.5.1.2 LOGIN.FRM - frmLOGIN - VB Form

- Purpose

Used to enter the server name and the login ID for the SACS user.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.5.1.3 MDIMAIN.FRM - mdlMain - VB Form

- Purpose

The base form for the other SACS forms.

- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.5.1.4 PASSWORD.FRM - frmPassword - VB Form

- Purpose
Allows the user to change the password for SACS login.
- Function
- Subordinates
- Dependencies
- Resources
- Processing
- Data

3.5.2 Database Design

3.5.2.1 Configuration Variables

The production SACS database has the following configuration variables set within Sybase.

Table 3. Sybase Database Configuration Variables

Name	min	max	config	run_value
recovery interval	1	32767	0	5
allow updates	0	1	0	0
user connections	5	2147483647	0	25
memory	3850	2147483647	53248	53248
open databases	5	2147483647	0	12
locks	5000	2147483647	0	5000
open objects	100	2147483647	500	500
procedure cache	1	99	20	20
fill factor	0	100	0	0
time slice	50	1000	0	100
database size	2	10000	0	2
tape retention	0	365	0	0
recovery flags	0	1	0	0
nested triggers	0	1	1	1
devices	4	256	25	25
remote access	0	1	1	1
remote logins	0	2147483647	2	2
remote sites	0	2147483647	2	2
remote connections	0	2147483647	5	5
pre-read packets	0	2147483647	3	3
upgrade version	0	2147483647	1002	1002
default sortorder id	0	255	50	50
default language	0	2147483647	0	0
language in cache	3	100	3	3
max online engines	1	32	1	1
min online engines	1	32	1	1
engine adjust interval	1	32	0	0
cpu flush	1	2147483647	200	200
i/o flush	1	2147483647	1000	1000
default character set id	0	255	1	1

stack size	20480	2147483647	0	28672
password expiration interval	0	32767	0	0
audit queue size	1	65535	100	100
additional netmem	0	2147483647	0	0
default network packet size	512	524288	0	512
maximum network packet size	512	524288	0	512
extent i/o buffers	0	2147483647	0	0
identity burning set factor	1	9999999	1	1

3.5.2.2 Data Dictionary - Tables

The data model for this database is contained in an ERwin diagram. Due to the difficulty in importing this diagram into this document, this diagram remains on paper. Four blank pages have been left so that the ERD can be inserted. Following the ERD is a description of each table.

VB_VERSION

VB_VERSION_TEXT: char(30)

U_REPORT_LOCKS

UID_LOCK: char(30)
DTTM_LOCK: datetime

VB_MESSAGE_OTD

GROUP_NAME: char(32)
MESSAGE: char(250)
START_DTTM: datetime
END_DTTM: datetime
EMP_ID: char(16)

ILL_VALUE CDX, clustered, allow dup
 NSTRUCTR_ID, ILL_DTTM

ILL_VALUE_NDX, non-clustered, unique
 LOW_WELL_NUMBER, ILL_DTTM, LOGGING_SEQUENCE_NO

BASELINES

BASELINE_ID: int
NSTRUCTR_ID: char(10)
SNSR_NO: int
START_DTTM: datetime
BASELINE_VALUE: float
BASELINE_DTTM: datetime
MEAS_NO: smallint
SLOPE_RATE: float
SLOPE_MEAS_NO: smallint
REL_UP_LMT: float
REL_LOW_LMT: float
ENTRY_DTTM: datetime
EMP_ID: char(16)
SNSR_TYPE_NAME: char(30)

ILL_VALUE

NSTRUCTR_ID: char(10)
LOW_WELL_NUMBER: char(6)
SNSR_TYPE_NAME: char(30)
ILL_DTTM: datetime
LOGGING_SEQUENCE_NO: smallint
OUTLIER_READING_COUNT: smallint
MARKER_METHOD: smallint
RAW_MARKER_DEPTH: float
DEPTH_CORRECTION: float
ILL_METHOD: smallint
ILL_CALC_DATE: datetime
RAW_ILL_VALUE: float
CORRECTED_ILL_VALUE: float
BASELINE_ILL_VALUE: float
MEAS_NO: smallint
DEVIATION: float
TOLERANCE: float
VALUE_IN_TOLERANCE: char(1)
PASS_FAIL_REVIEW: char(1)
REVIEWED_EMP_ID: char(7)
ILL_COMMENT: varchar(60)

TANK GROUP	
TANK_GROUP_NAME:	char(30)
TANK_GROUP_DESC:	text

NAMED STRUCTURE RECUR	
HOLDER:	char(10)
HOLDEE:	char(10)

NS COMMENT	
NS_COMMENT_ID:	numeric
NS_COMMENT_TEXT:	varchar(250)
NS_COMMENT_DTTM:	datetime
EMP_ID:	char(16)
NS_ENTRY_DTTM:	datetime
NSTRUCTR_ID:	char(10)

TANK IN GROUP	
TANK_GROUP_NAME:	char(30)
NSTRUCTR_ID:	char(10)
GROUP_START_DTTM:	datetime
GROUP_END_DTTM:	datetime

NAMED STRUCTURE	
NSTRUCTR_ID:	char(10)
NSTRUCTTYP_NO:	tinyint
NSTRUCTR_TYPE_NAME:	char(20)
NSTRUCTR_NAME:	char(30)
NSTRUCTR_COMN_NAME:	char(20)

NS COMMENT AUDIT	
NS_COMMENT_ID:	numeric
NSTRUCTR_ID:	char(10)
NS_COMMENT_TEXT:	varchar(250)
NS_COMMENT_DTTM:	datetime
EMP_ID:	char(16)
NS_ENTRY_DTTM:	datetime
NS_COM_AUD_ACTION:	char(1)

RDG COLCTN SYS SNSR TYPE	
RCSYS_NO:	smallint
SNSR_TYPE_NAME:	char(30)
NSTRUCTTYP_NO:	tinyint
NSTRUCTR_NO:	char(10)
FSAMPLSPCTY_NO:	tinyint
FSAMPLSPC_NO:	smallint

FIXED SAMPLE SPACE	
FSS_ID:	numeric
NSTRUCTTYP_NO:	tinyint
NSTRUCTR_ID:	char(10)
FSS_TYPE_NO:	smallint
FSS_TYPE_NAME:	char(30)
FSAMPLSPC_NO:	smallint
FSS_NAME:	char(30)
COMN_NAME:	char(20)

FSS FREQUENCY	
FSS_ID:	numeric
FREQ_NO:	smallint
SNSR_RDG_GROUP:	char(15)

FREQUENCY DEFN	
FREQ_NO:	smallint
FREQ_NAME:	char(30)
LIMIT:	real
MEAS_NO:	smallint
WARNING_VALUE:	real
WARN_MEAS_NO:	smallint
EXTENSION_VALUE:	real
EXTENSION_MEAS_NO:	smallint

FREQ VIO REPORT	
SNSR_RDG_GROUP:	char(15)
NSTRUCTR_ID:	char(10)
FSS_TYPE_NO:	smallint
FSAMPLSPC_NO:	smallint
FREQ_NO:	smallint
RPT_RUN_DTTM:	datetime
COMN_NAME:	char(20)
SENSOR_NAME:	char(30)
LAST_SAMPLE_DTTM:	datetime
DAYS_SINCE_GOOD_RDG:	smallint
SUSP_DATA_STAT_CODE:	char(1)
DUE_LIMIT_DTTM:	datetime
DUE WARNING_DTTM:	datetime
DUE_EXTENDED_DTTM:	datetime
LIMIT_VIOLATED:	char(1)
WARNING_VIOLATED:	char(1)
EXTENDED_VIOLATED:	char(1)

READING COLCTN SYSTEM	
RCSYS_NO:	smallint
ABBREVD_NAME:	char(10)
RCSYS_NAME:	char(50)

FSS SENSOR	
FSS_ID:	int
SNSR_NO:	int
INSTALL_DT:	datetime
REMOVAL_DT:	datetime
IN_FROM_BOTTOM:	float

SURV ALERTS ILL

RDG_DTTM: datetime
 NSTRUCTR_ID: char(10)
 LOW_WELL_NO: char(6)
 SNSR_TYPE_NAME: char(30)
 RDG_VALUE: float
 BASELINE_CURRENT: float
 PASS_FAIL_REVIEW: char(1)
 REL_UP_LMT: float
 REL_LOW_LMT: float
 GOOD_VALUES: int
 INCREASE_50: int
 DECREASE_50: int
 INCREASE_EX: int
 DECREASE_EX: int
 RUN_DTTM: datetime

SURV ALERTS TPTR

RDG_DTTM: datetime
 NSTRUCTR_ID: char(10)
 COMN_NAME: char(20)
 COLCTN_IDX: int
 RCSYS_ABBREVD_NAME: char(10)
 RDG_VALUE: float
 BASELINE_CURRENT: float
 QUALITY_INDICATOR: char(1)
 REL_UP_LMT: float
 REL_LOW_LMT: float
 GOOD_VALUES: int
 INCREASE_50: int
 DECREASE_50: int
 INCREASE_EX: int
 DECREASE_EX: int
 RUN_DTTM: datetime

SURV ALERTS SLVL

RDG_DTTM: datetime
 NSTRUCTR_ID: char(10)
 SNSR_TYPE_NAME: char(30)
 RDG_VALUE: float
 BASELINE_CURRENT: float
 QUALITY_INDICATOR: char(1)
 REL_UP_LMT: float
 REL_LOW_LMT: float
 GOOD_VALUES: int
 INCREASE_50: int
 DECREASE_50: int
 INCREASE_EX: int
 DECREASE_EX: int
 RUN_DTTM: datetime

U SNSR PTS RET PARM

EMP_ID: char(16)
 RCSYS_NO: smallint
 SNSR_NO: int
 BEG_DTTM: datetime
 END_DTTM: datetime
 SNSR_TYPE_NAME: char(30)

U TEMPERATURE REPORTS

TANK: char(10)
 DATA_SOURCE: char(10)
 RISER: char(20)
 THERMOCOUPLE: tinyint
 TEMPERATURE: float
 UNITS: char(1)
 DATE: datetime
 NO_SLVL_FOUND: char(1)

SURVEILLANCE ALERTS

RDG_DTTM: datetime
 NSTRUCTR_ID: char(10)
 COMN_NAME: char(20)
 COLCTN_IDX: int
 RDG_VALUE: float
 BASELINE_VALUE: real
 QUALITY_INDICATOR: char(1)
 GOOD_VALUES: char(1)
 INCREASE_50: char(1)
 DECREASE_50: char(1)
 INCREASE_EX: char(1)
 DECREASE_EX: char(1)
 RUN_DTTM: datetime

CODE

CODE: char(20)
 COLUMN_SQL_NAME: char(30)
 CODE_TYPE: char(40)
 CODE_NAME: char(40)

SLVL READING AUDIT

SLVL_ID: numeric
 RCSYS_NO: smallint
 SNSR_NO: int
 SLVL_DTTM: datetime
 NSTRUCTR_ID: char(10)
 SLVL_QTY: float
 MEAS_NO: smallint
 DATA_QUAL_STAT_CODE: char(1)
 SUSP_DATA_STAT_CODE: char(1)
 DATA_ENTRY_DTTM: datetime
 EMP_ID: char(16)
 REVIEWED_EMP_ID: char(7)
 AUDIT_CODE: char(1)

TPTR READING AUDIT

TPTR_ID: numeric
 RCSYS_NO: smallint
 SNSR_NO: int
 TPTR_DTTM: datetime
 NSTRUCTR_ID: char(10)
 MRDCNAM_NO: smallint
 TPTR_QTY: float
 MEAS_NO: smallint
 DATA_QUAL_STAT_CODE: char(1)
 SUSP_DATA_STAT_CODE: char(1)
 DATA_ENTRY_DTTM: datetime
 EMP_ID: char(16)
 REVIEWED_EMP_ID: char(7)
 AUDIT_CODE: char(1)

BASELINES - Entity Definition: This table contains the baseline values and limits for readings in the SACS database. Values are maintained by the customer.

BASELINE_ID (PK)
 NSTRUCTR_ID (FK)
 SNSR_NO (FK)
 START_DTTM
 BASELINE_VALUE
 BASELINE_DTTM
 MEAS_NO (FK)
 SLOPE_RATE
 SLOPE_MEAS_NO
 REL_UP_LMT
 REL_LOW_LMT
 ENTRY_DTTM
 EMP_ID
 SNSR_TYPE_NAME

COLLECTION - Entity Definition: This is a number of sensors that are considered as a group. More commonly known as a thermocouple tree.

COLCTN_NO (PK)
 COLCTN_NAME

FIXED_SAMPLE_SPACE - Entity Definition: This is a bounded volume where samples are taken. Examples are tanks and risers within tanks. Sensors take their samples within volumes.

FSS_ID (PK)
 NSTRUCTTYP_NO
 NSTRUCTR_ID (FK)
 FSS_TYPE_NO
 FSS_TYPE_NAME
 FSAMPLSPC_NO
 FSS_NAME
 COMN_NAME

FREQUENCY_DEFN - Entity Definition: This table holds the different frequency definitions. These definitions are used to determine whether a fixed sample space has been sampled frequently enough to meet requirements. Values are maintained by the customer.

FREQ_NO (PK)
 FREQ_NAME
 LIMIT
 MEAS_NO
 WARNING_VALUE
 WARN_MEAS_NO
 EXTENSION_VALUE
 EXTENSION_MEAS_NO

FREQ_VIO_REPORT - Entity Definition: This is a temporary table that hold the frequency violation report. It is emptied and filled by the lp_MakeFreqVioRpt stored procedure.

SNSR_RDG_GROUP (PK)
 NSTRUCTR_ID (PK) (FK)
 FSS_TYPE_NO (PK)
 FSAMPLSPC_NO (PK)
 FREQ_NO (PK) (FK)
 RPT_RUN_DTTM
 COMN_NAME
 SENSOR_NAME
 LAST_SAMPLE_DTTM
 DAYS_SINCE_GOOD_RDG
 SUSP_DATA_STAT_CODE
 DUE_LIMIT_DTTM
 DUE_WARNING_DTTM
 DUE_EXTENDED_DTTM
 LIMIT_VIOLATED
 WARNING_VIOLATED
 EXTENDED_VIOLATED

FSS_FREQUENCY - Entity Definition: This table relate a frequency to a fixed sample space and a type of sensor. It is part of the frequency violation system.

FSS_ID (PK) (FK)
 FREQ_NO (PK) (FK)
 SNSR_RDG_GROUP

FSS_SENSOR - Entity Definition: This table relates a sensor to a fixed sample space. A sensor placed in a fixed sample space has an entry in this table.

FSS_ID (PK)
 SNSR_NO (PK) (FK)
 INSTALL_DT
 REMOVAL_DT
 IN_FROM_BOTTOM

ILL_VALUE - Entity Definition: This table holds ILL readings received from CASS. All of the computation and evaluation for these readings are done on CASS.

NSTRUCTR_ID (FK)
 LOW_WELL_NUMBER
 SNSR_TYPE_NAME
 ILL_DTTM
 LOGGING_SEQUENCE_NO
 OUTLIER_READING_COUNT
 MARKER_METHOD
 RAW_MARKER_DEPTH
 DEPTH_CORRECTION
 ILL_METHOD
 ILL_CALC_DATE
 RAW_ILL_VALUE
 CORRECTED_ILL_VALUE
 BASELINE_ILL_VALUE
 MEAS_NO (FK)
 DEVIATION
 TOLERANCE

VALUE_IN_TOLERANCE
 PASS_FAIL_REVIEW
 REVIEWED_EMP_ID
 ILL_COMMENT

MAN_RDG_DEV_COMN_NAME - Entity Definition: This table provides a translation between the names commonly used for hand-held temperature reading devices and the formal names of these devices. As a practical matter this table gets update from the temperature input form; the reading device table does not get updated.

MRDCNAM_NO (PK)
 MRDCNAM_NAME
 RDGDEV_NO (FK)

MEASURE - Entity Definition: This table contains the different measures understood by SACS.

MEAS_NO (PK)
 MEAS_CODE
 MEASTYP_NO
 MEASTYP_CODE
 MEAS_NAME
 TYPE_NAME

MEAS_BASE_CONVN - Entity Definition: This table contains pointers and conversion factors that allow a reading taken in one unit of measure to be converted to another unit of measure.

MEAS_NO (PK) (FK)
 MEAS_CODE
 BASE_MEAS_NO (FK)
 BASE_MEAS_CODE
 RATIO
 OFFSET

NAMED_STRUCTURE - Entity Definition: This table contains all of the named structures known to SACS. As of Rev 03.05.01 this means it contains tanks, catch tanks, and farms.

NSTRUCTR_ID (PK)
 NSTRUCTYP_NO
 NSTRUCTR_TYPE_NAME
 NSTRUCTR_NAME
 NSTRUCTR_COMN_NAME

NAMED_STRUCTURE_RECUR - Entity Definition: This table provides a way to show that one named structure is contained within another. Used because we have tanks and farms both in the named structure table.

HOLDER (PK) (FK)
 HOLDEE (PK) (FK)

NS_COMMENT - Entity Definition: This table contains comments that the customer has made on named structures.

NS_COMMENT_ID (PK)
 NS_COMMENT_TEXT
 NS_COMMENT_DTTM

EMP_ID
 NS_ENTRY_DTTM
 NSTRUCTR_ID (FK)

NS_COMMENT_AUDIT - Entity Definition: This table is the audit trail for the NS_COMMENT table. Changes made to the NS_COMMENT table are recorded here by triggers in the NS_COMMENT table.

NS_COMMENT_ID (PK) (FK)
 NSTRUCTR_ID (FK)
 NS_COMMENT_TEXT
 NS_COMMENT_DTTM
 EMP_ID
 NS_ENTRY_DTTM
 NS_COM_AUD_ACTION

RDGDEV_VALID_SENSOR - Entity Definition: This table contains a list of the reading devices that are valid for a given sensor type. For each reading device there is a limit on the valid values that it may return for that type of sensor. For example, a certain type of thermocouple reader may only return valid readings in the 100-450 degree range for a J-type thermocouple. Any reading outside this range would be suspect.

RDG_DEVICE_TYPE_NAME
 SNSR_TYPE_NAME
 LWR_LMT_QTY
 LWR_LMT_MEAS_NO
 LWR_LMT_MEAS_CODE
 UPR_LMT_QTY
 UPR_LMT_MEAS_NO
 UPR_LMT_MEAS_CODE

RDG_COLCTN_SYS_SENSOR - Entity Definition: This table relates our internal sensor names to how the sensor is known by an external data collection system. A sensor may have several names depending on who is collecting the data; this table holds them all. It may be possible to pare columns from this table and reduce it to RCSYS_NO, SNSR_NO, and TAG_NAME some time in the future.

RCSYS_NO (PK) (FK)
 SNSR_NO (PK) (FK)
 NSTRUCTTYP_NO
 NSTRUCTR_ID (FK)
 FSMPLSPCTY_NO
 FSAMPLSPC_NO
 COLCTN_NO
 COLCTN_SNSR_IDX
 TAG_NAME

RDG_COLCTN_SYS_SNSR_TYPE - Entity Definition: This table holds a list of the different reading collection systems and the types of sensors that each system has in a given fixed sample space. Used as part of the frequency violation report. May benefit from replacing the 4-column-join with an FSS_ID.

RCSYS_NO (PK) (FK)
 SNSR_TYPE_NAME (PK)
 NSTRUCTTYP_NO

NSTRUCTR_NO
FSMPLSPCTY_NO
FSAMPLSPC_NO

RDG_DEVICE - Entity Definition: This table holds the "real" names of the hand-held reading devices used to collect temperature data.

RDGDEV_NO (PK)
RDGDEV_NAME
RDGDEVTYP_NO
RDGDEV_TYPE_NAME

READING_COLCTN_SYSTEM - Entity Definition: This table contains the name of external data collection systems known to SACS.

RCSYS_NO (PK)
ABBREVD_NAME
RCSYS_NAME

SENSOR - Entity Definition: This table contains a list of all the sensors known to SACS. A sensor is a physical device in the field.

SNSR_NO (PK)
SNSR_IS_REFERENCE_CODE
SNSR_TYPE_NAME
COLCTN_NO
COLCTN_IDX
MEAS_TYPE_NO

SENSOR_TYPE - Entity Definition: This table contains a list of the different types of sensors known to SACS.

SNSR_TYPE_CD (PK)
SNSR_TYPE_NAME
SNSR_RDG_GROUP

SLVL_READING - Entity Definition: This table contains the surface level readings for SACS.

SLVL_ID (PK)
RCSYS_NO (FK)
SNSR_NO (FK)
SLVL_DTTM
NSTRUCTR_ID (FK)
SLVL_QTY
MEAS_NO (FK)
DATA_QUAL_STAT_CODE
SUSP_DATA_STAT_CODE
DATA_ENTRY_DTTM
EMP_ID
SLVL_COMMENT

SLVL_READING_AUDIT - Entity Definition: This table is the audit trail for the SLVL_READING table. Triggers in SLVL_READING update the audit trail when records are modified or deleted.

SLVL_ID (PK)
RCSYS_NO (PK)

SNSR_NO (PK)
 SLVL_DTTM (PK)
 NSTRUCTR_ID
 SLVL_QTY
 MEAS_NO
 DATA_QUAL_STAT_CODE
 SUSP_DATA_STAT_CODE
 DATA_ENTRY_DTTM
 EMP_ID
 REVIEWED_EMP_ID
 AUDIT_CODE

SURV_ALERTS_ILL - Entity Definition: This table holds the surveillance alerts report for the ILL readings. This table holds data only temporarily.

RDG_DTTM
 NSTRUCTR_ID
 LOW_WELL_NO
 SNSR_TYPE_NAME
 RDG_VALUE
 BASELINE_CURRENT
 PASS_FAIL_REVIEW
 REL_UP_LMT
 REL_LOW_LMT
 GOOD_VALUES
 INCREASE_50
 DECREASE_50
 INCREASE_EX
 DECREASE_EX
 RUN_DTTM

SURV_ALERTS_SLVL - Entity Definition: This table holds the surveillance alerts report for the surface level readings. This table holds data only temporarily.

RDG_DTTM
 NSTRUCTR_ID
 SNSR_TYPE_NAME
 RDG_VALUE
 BASELINE_CURRENT
 QUALITY_INDICATOR
 REL_UP_LMT
 REL_LOW_LMT
 GOOD_VALUES
 INCREASE_50
 DECREASE_50
 INCREASE_EX
 DECREASE_EX
 RUN_DTTM

SURV_ALERTS_TPTR - Entity Definition: This table holds the surveillance alerts report for the temperature readings. This table holds data only temporarily.

RDG_DTTM
 NSTRUCTR_ID

COMN_NAME
 COLCTN_IDX
 RCSYS_ABBREVD_NAME
 RDG_VALUE
 BASELINE_CURRENT
 QUALITY_INDICATOR
 REL_UP_LMT
 REL_LOW_LMT
 GOOD_VALUES
 INCREASE_50
 DECREASE_50
 INCREASE_EX
 DECREASE_EX
 RUN_DTTM

S_COMMENT - Entity Definition: This table holds comments that relate to a sensor.

S_COMMENT_ID (PK)
 SNSR_NO (FK)
 S_COMMENT_TEXT
 S_COMMENT_DTTM
 EMP_ID
 S_ENTRY_DTTM

S_COMMENT_AUDIT - Entity Definition: This table is the audit trail for the S_COMMENT table. Triggers in the S_COMMENT table update the audit trail whenever records are modified or deleted.

S_COMMENT_ID (PK) (FK)
 SNSR_NO (FK)
 S_COMMENT_TEXT
 S_COMMENT_DTTM
 EMP_ID
 S_ENTRY_DTTM
 S_COM_AUD_ACTION

TANK_GROUP - Entity Definition: This table contains a list of the tank groups. Tank groups are used for the maximum temperature report. Values are maintained by the customer.

TANK_GROUP_NAME (PK)
 TANK_GROUP_DESC

TANK_IN_GROUP - Entity Definition: This table relates a waste storage tank to a tank group. Since a tank may join, leave, and re-join a group the date and time for each joining is stored. Values are maintained by the customer.

TANK_GROUP_NAME (PK) (FK)
 NSTRUCTR_ID (PK) (FK)
 GROUP_START_DTTM (PK)
 GROUP_END_DTTM

TPTR_READING - Entity Definition: This table holds the temperature readings for the SACS system.

TPTR_ID (PK)
 RCSYS_NO (FK)

SNSR_NO (FK)
 TPTR_DTTM
 NSTRUCTR_ID (FK)
 MRDCNAM_NO (FK)
 TPTR_QTY
 MEAS_NO (FK)
 DATA_QUAL_STAT_CODE
 SUSP_DATA_STAT_CODE
 DATA_ENTRY_DTTM
 EMP_ID
 TPTR_COMMENT

TPTR_READING_AUDIT - Entity Definition: This table is the audit trail for the TPTR_READING table. Triggers in the TPTR_READING table update the audit trail whenever rows are modified or deleted.

TPTR_ID (PK)
 RCSYS_NO (PK)
 SNSR_NO (PK)
 TPTR_DTTM (PK)
 NSTRUCTR_ID
 MRDCNAM_NO
 TPTR_QTY
 MEAS_NO
 DATA_QUAL_STAT_CODE
 SUSP_DATA_STAT_CODE
 DATA_ENTRY_DTTM
 EMP_ID
 REVIEWED_EMP_ID
 AUDIT_CODE

U_REPORT_LOCKS - Entity Definition: This table contains the locks for different reports that the custome generates. Rows are ephemeral.

UID_LOCK
 DTTM_LOCK

U_SNSR_PTS_RET_PARM - Entity Definition: The client fills this table with parameters about the data that it wants to retrieve. Rows are ephemeral.

EMP_ID (PK)
 RCSYS_NO (PK) (FK)
 SNSR_NO (PK) (FK)
 BEG_DTTM
 END_DTTM
 SNSR_TYPE_NAME

U_TEMPERATURE_REPORTS - Entity Definition: This table holds the current temperature report. Its main use is to hold rows that Crystal Reports can use to make the maximum temperature report. Rows are ephemeral.

TANK
 DATA_SOURCE
 RISER
 THERMOCOUPLE

TEMPERATURE
UNITS
DATE
NO_SLVL_FOUND

VB_MESSAGE_OTD - Entity Definition: This table contains the message that will be displayed to the user upon login.

GROUP_NAME
MESSAGE
START_DTTM
END_DTTM
EMP_ID

VB_VERSION - Entity Definition: This table is used to coordinate the client version with the database version.

VB_VERSION_TEXT

3.5.2.3 Triggers on Tables

Table 4. Triggers on Tables

Table Name	Kind of Trigger	Trigger Name	Purpose
FIXED_SAMPLE_SPACE	Insert	FIXED_SAMPLE_SPACE_INS	Sets the next FSS_ID number when a row is added.
FREQUENCY_DEFN	Delete	ID_FREQUENCY_DEFN	Cascading delete - when a frequency definition is deleted all the rows that use this definition in the FSS_SAMPLE_FREQ and FREQ_VIO_REPORT are deleted.
FSS_FREQUENCY	Insert	FSS_FREQUENCY_INS	Maintains referential integrity with FREQUENCY_DEFN and FIXED_SAMPLE_SPACE.
FSS_FREQUENCY	Update	FSS_FREQUENCY_UPD	Maintains referential integrity with FREQUENCY_DEFN and FIXED_SAMPLE_SPACE.
ILL_VALUE	Insert	Ill_value_insert_trig	Maintains referential integrity with NAMED_STRUCTURE
NS_COMMENT	Delete	NS_COMMENT_DEL	Maintains audit trail in NS_COMMENT_AUDIT.
NS_COMMENT	Insert	NS_COMMENT_INS	Maintains referential integrity with NAMED_STRUCTURE. Inserts date and EMP_ID of person who inserted the row.
NS_COMMENT	Update	NS_COMMENT_UPD	Maintains referential integrity with NAMED_STRUCTURE. Maintains audit trail in NS_COMMENT_AUDIT.

Table Name	Kind of Trigger	Trigger Name	Purpose
S_COMMENT	Delete	S_COMMENT_DEL	Maintains audit trail in S_COMMENT_AUDIT.
S_COMMENT	Insert	S_COMMENT_INS	Maintains referential integrity with SENSOR. Inserts date and EMP_ID of person who inserted the row.
S_COMMENT	Update	S_COMMENT_UPD	Maintains referential integrity with SENSOR. Maintains audit trail in S_COMMENT_AUDIT.
SENSOR	Delete	SENSOR_DEL	Disallows delete if sensor is in use in SLVL_READING, TPTR_READING, BASELINES, S_COMMENT. Cascades delete into RDG_COLCTN_SYS_SENSOR and FSS_SENSOR tables.
SENSOR	Insert	SENSOR_INS	Maintains referential integrity with COLLECTION.
SENSOR	Update	SENSOR_UPD	Disallows updates of SNSR_NO column. Maintains referential integrity with COLLECTION.
SLVL_READING	Delete	SLVL_READING_DEL	Maintains audit trail in SLVL_READING_AUDIT.
SLVL_READING	Insert	SLVL_READING_INS	Maintains referential integrity with SENSOR, READING_COLCTN_SYSTEM, NAMED_STRUCTURE. Inserts date and EMP_ID of person who inserted the row. Sets the SUSP_DATA_STAT_CODE = 'S' for values < 1.
SLVL_READING	Update	SLVL_READING_UPD	Maintains referential integrity with SENSOR, READING_COLCTN_SYSTEM, NAMED_STRUCTURE. Maintains audit trail in SLVL_READING_AUDIT.
TANK_GROUP	Delete	ID_TANK_GROUP	Cascades delete to all related rows in TANK_IN_GROUP table.
TANK_GROUP	Update	IU_TANK_GROUP	Disallows update to TANK_GROUP_NAME if related TANK_IN_GROUP rows exist.

Table Name	Kind of Trigger	Trigger Name	Purpose
TANK_IN_GROUP	Insert	ti_TANK_IN_GROUP	Maintains referential integrity with NAMED_STRUCTURE and TANK_GROUP. Disallows insert if an identical tank group exists with a NULL end-date.
TANK_IN_GROUP	Update	tu_TANK_IN_GROUP	Maintains referential integrity with NAMED_STRUCTURE and TANK_GROUP.
TPTR_READING	Delete	TPTR_READING_DEL	Maintains audit trail in TPTR_READING_AUDIT.
TPTR_READING	Insert	TPTR_READING_INS	Maintains referential integrity with SENSOR, READING_COLCTN_SYSTEM, NAMED_STRUCTURE, MAN_RDG_DEV_COMN_NAME. Inserts date and EMP_ID of person who inserted the row. Sets the SUSP_DATA_STAT_CODE = 'S' for values < 1.
TPTR_READING	Update	TPTR_READING_UPD	Maintains referential integrity with SENSOR, READING_COLCTN_SYSTEM, NAMED_STRUCTURE, MAN_RDG_DEV_COMN_NAME. Maintains audit trail in TPTR_READING_AUDIT.

3.5.2.4 Data Dictionary - Columns

For each column this data dictionary lists:

- The name of the column
- A definition
- The table(s) where used
- The type of column
- The default procedure for this column
- The rule or domain procedure for this column
- The source of data for this column.

Optional lines are:

- Alias - the same data element under a different name in other tables.
- Values - Where the number of possible values is limited, a list of the values and their meanings is given.

ABBREVD_NAME

Description: A short name for a reading collection system.

Used in: READING_COLCTN_SYSTEM

Type: char 10
 Default: NULL
 Rule: NULL
 Source: SACS Developers

AUDIT_CODE

Description: A code that describes the action that caused a row to be added to the audit trail.
 Used in: SLVL_READING_AUDIT, TPTR_READING_AUDIT
 Type: char 1
 Default: NULL
 Rule: NULL
 Values: U - Update. A change was made to the record.
 D - Delete. The record was deleted.
 Source: Triggers for update and delete on the readings tables.

BASELINE_CURRENT

Description: The value of a baseline, sloping or flat, at the date-time contained in the row of the table.
 Used in: SURV_ALERTS_ILL, SURV_ALERTS_SLVL, SURV_ALERTS_TPTR
 Type: float
 Default: NULL
 Rule: NULL
 Source: Calculated from the BASELINE table (BASELINE_VALUE, BASELINE_DTTM, SLOPE_RATE) and reading date-time values.

BASELINE_DTTM

Description: The date-time that the offset of the sloping baseline = 0.
 Used in: BASELINES
 Type: datetime
 Default: NULL
 Rule: NULL
 Source: Customer input.

BASELINE_ID

Description: Unique key used internally in the database to identify baselines.
 Used in: BASELINES
 Type: int
 Default: NULL
 Rule: NULL
 Source: Generated by database as an identify data type.

BASELINE_ILL_VALUE

Description: The baseline in effect for the ILL reading.
 Used in: ILL_VALUE
 Type: float
 Default: NULL
 Rule: NULL
 Source: Received from CASS as part of the ILL reading.

BASELINE_VALUE

Description: The value of the baseline at the BASELINE_DTTM.

Used in: BASELINES, SURVEILLANCE_ALERTS

Type: float

Default: NULL

Rule: NULL

Source: Customer input.

BASE_MEAS_CODE

Description: The base measurement code. When performing conversions using MEAS_CONV_N_V the source measure is converted into units of the base measure, then converted to the target measure. This way adding a new measure is a linear increment of work instead of a squared increment.

Used in: MEAS_BASE_CONV_N

Type: char 10

Default: NULL

Rule: NULL

Source: SACS Developers

BASE_MEAS_NO

Description: The base measurement number. When performing conversions using MEAS_CONV_N_V the source measure is converted into units of the base measure, then converted to the target measure. This way adding a new measure is a linear increment of work instead of a squared increment.

Used in: MEAS_BASE_CONV_N

Type: smallint

Default: NULL

Rule: NULL

Source: SACS Developers

BEG_DTTM

Description: The starting date-time to retrieve data.

Used in: U_SNSR_PTS_RET_PARM

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input.

COLCTN_IDX

Description: The offset of the sensor within a tree or probe. Usually sensors on an instrument tree are numbered from the bottom starting with 1. Occasionally (like AP-farm) they are numbered from the top.

Used in: SENSOR, SURV_ALERTS_TPTR, SURVEILLANCE_ALERTS

Aliases: RDG_COLCTN_SYS_SENSORCOLCTN_SNSR_IDX

Type: int

Default: NULL

Rule: NULL

Source: Customer configuration information.

COLCTN_NAME

Description: The name of a collection. Since the customer hasn't named most collection, this is used as a holder for historical names of trees. This information is useful when converting old data.

Used in: COLLECTION

Type: char 30

Default: NULL

Rule: NULL

Source: Customer configuration information.

COLCTN_NO

Description: A unique number used within the database to identify a collection.

Used in: COLLECTION, RDG_COLCTN_SYS_SENSOR, SENSOR

Type: smallint

Default: NULL

Rule: NULL

Source: SACS Developers

COLCTN_SNSR_IDX

Description: The offset of the sensor within a tree or probe.

Used in: RDG_COLCTN_SYS_SENSOR

Aliases: COLCTN_IDX in SENSOR, SURV_ALERTS_TPTR, SURVEILLANCE_ALERTS

Type: int

Default: NULL

Rule: NULL

Source: Customer configuration information.

COMMENT_DTTM

Description: The date-time that a comment refers to. A copy of the date-times associated with comments in the S_COMMENT, NS_COMMENT, TPTR_READING, and SLVL_READING tables.

Used in: U_COMMENT_REPORT

Type: datetime

Default: NULL

Rule: NULL

Source: Copied from other tables.

COMMENT_TEXT

Description: The text of the comment. A copy of the text for comments in the S_COMMENT, NS_COMMENT, TPTR_READING, and SLVL_READING tables.

Used in: U_COMMENT_REPORT

Type: char 250

Default: NULL

Rule: NULL

Source: Copied from other tables.

COMMENT_TYPE

Description: The source of the comment for this row of the table.

Used in: U_COMMENT_REPORT

Type: char 20

Default: NULL

Rule: NULL

Values: "Reading Comments"

"Tank Comments"

"Sensor Comments"

Source: Code within Ip_GetRDGComments.

COMN_NAME

Description: The common name of a fixed sample space. This is how the surveillance customer has named a fixed sample space. The official value for this name is kept in the FIXED_SAMPLE_SPACE table.

Used in: FIXED_SAMPLE_SPACE, FREQ_VIO_REPORT, SURV_ALERTS_TPTR, SURVEILLANCE_ALERTS

Type: char 20

Default: NULL

Rule: NULL

Source: Customer configuration information.

CORRECTED_ILL_VALUE

Description: The ILL value that has been adjusted using customer parameters. This is the official ILL value, calculated on CASS and sent to SACS.

Used in: ILL_VALUE

Type: float

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

DATA_ENTRY_DTTM

Description: The date-time that data were entered into the database or were modified.

Used in: SLVL_READING, SLVL_READING_AUDIT, TPTR_READING, TPTR_READING_AUDIT

Type: datetime

Default: NULL

Rule: NULL

Source: Trigger in database table.

DATA_QUAL_STAT_CODE

Description: A code used to show whether the reading device could actually have taken a reading with this value. For example, a reading of -500 deg. F is not physically possible. That reading would have a value of 'I' for invalid.

Used in: SLVL_READING_AUDIT, TPTR_READING_AUDIT

Type: char 1

Default: NULL

Rule: NULL

Values: 'V' - valid reading.

'I' - invalid reading.

Source: Copied from SLVL_READING or TPTR_READING table.

DATA_QUAL_STAT_CODE

Description: A code used to show whether the reading device could actually have taken a reading with this value. For example, a reading of -500 deg. F is not physically possible. That reading would have a value of 'I' for invalid.

Used in: SLVL_READING, TPTR_READING

Type: char 1

Default: rdgs_data_qual_stat_def

Rule: rdgs_data_qual_stat_rule

Values: 'V' - valid reading.

'I' - invalid reading.

Source: Customer input or, more usually, the default code.

DATA_SOURCE

Description: The reading collection system that provided this row in the table.

Used in: U_TEMPERATURE_REPORTS

Type: char 10

Default: NULL

Rule: NULL

Values: 'TMACS'

Source: Retrieved using clues in the readings tables.

DATE

Description: The date-time of the reading in this row of the report.

Used in: U_TEMPERATURE_REPORTS

Type: datetime

Default: NULL

Rule: NULL

Source: Copied from the readings tables.

DAYS_SINCE_GOOD_RDG

Description: The number of days since the last good reading for this type of sensor in this tank.

Used in: FREQ_VIO_REPORT

Type: int

Default: NULL

Rule: NULL

Source: Calculated from reading dates in the readings tables and the report run date.

DECREASE_50

Description: A flag that says that the reading for this row is more than 50% of the way from the baseline to the lower surveillance limit but not lower than the surveillance limit. Used by the Surveillance Alert Report to select only the rows wanted.

Used in: SURV_ALERTS_TPTR, SURV_ALERTS_ILL, SURV_ALERTS_SLVL, SURVEILLANCE_ALERTS

Type: int

Default: NULL

Rule: NULL

Values: NULL, not NULL

Source: Calculated from the BASELINE_CURRENT, REL_LOW_LMT, and the reading values.

DECREASE_EX

Description: A flag that says that the reading for this row is below the lower surveillance limit. Used by the Surveillance Alert Report to select only the rows wanted.

Used in: SURV_ALERTS_SLVL, SURV_ALERTS_ILL, SURV_ALERTS_TPTR, SURVEILLANCE_ALERTS

Type: int

Default: NULL

Rule: NULL

Values: NULL, not NULL

Source: Calculated from the BASELINE_CURRENT, REL_LOW_LMT, and the reading values.

DEPTH_CORRECTION

Description: The amount of correction to apply to the RAW_ILL_VALUE to calculate the CORRECTED_ILL_VALUE.

Used in: ILL_VALUE

Type: float

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

DEVIATION

Description: The difference between the CORRECTED_ILL_VALUE and the BASELINE_ILL_VALUE.

Used in: ILL_VALUE

Type: float

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

DUE_EXTENDED_DTTM

Description: The frequency definition contains an EXTENSION_VALUE. The DUE_EXTENDED_DTTM is (the date-time of the last good reading for a fixed sample space for that type of sensor) + (EXTENSION_VALUE).

Used in: FREQ_VIO_REPORT

Type: datetime

Default: NULL

Rule: NULL

Source: Calculated from the date-time of the last good reading and the EXTENSION_VALUE for the fixed sample space.

DUE_LIMIT_DTTM

Description: The frequency definition contains an LIMIT. The DUE_LIMIT_DTTM is (the date-time of the last good reading for a fixed sample space for that type of sensor) + (LIMIT).

Used in: FREQ_VIO_REPORT

Type: datetime

Default: NULL

Rule: NULL

Source: Calculated from the date-time of the last good reading and the LIMIT for the fixed sample space.

DUE_WARNING_DTTM

Description: The frequency definition contains a WARNING_VALUE. The DUE_WARNING_DTTM is (the date-time of the last good reading for a fixed sample space for that type of sensor) + (WARNING_VALUE).

Used in: FREQ_VIO_REPORT

Type: NULL NULL

Default: NULL

Rule: NULL

Source: Calculated from the date-time of the last good reading and the WARNING_VALUE for the fixed sample space.

EMP_ID

Description: The ID of the employee. Based on the employee ID used by the Hanford contractors.

Used in: U_SNSR_PTS_RET_PARM, VB_MESSAGE_OTD, BASELINES, TPTR_READING_AUDIT, SLVL_READING_AUDIT, NS_COMMENT_AUDIT, S_COMMENT_AUDIT, TPTR_READING, SLVL_READING, NS_COMMENT, S_COMMENT

Type: char 30

Default: NULL

Rule: NULL

Source: PeopleCore.

END_DTTM

Description: The ending date-time to retrieve data for.

Used in: U_SNSR_PTS_RET_PARM

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input.

END_DTTM

Description: The last date-time to display this message.

Used in: VB_MESSAGE_OTD

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input when defining a message to display.

ENTRY_DTTM

Description: The date-time that the baseline was entered into the system.

Used in: BASELINES

Type: datetime

Default: NULL

Rule: NULL

Source: Provided by the client software.

EXTENDED_VIOLATED

Description: A flag used to indicate that a sample-type - fixed-sample-space extended frequency has been violated. Used by the report to select the rows that it wants.

Used in: FREQ_VIO_REPORT

Type: char 1

Default: NULL

Rule: NULL

Values: 'Y' - limit has been violated.

' ' - limit has not been violated.

Source: Calculated by lp_MakeFreqVioRpt.

EXTENSION_MEAS_NO

Description: The unit of measure for the extended frequency. Ref. The MEASURE table for the details.

Used in: FREQUENCY_DEFN

Type: smallint

Default: NULL

Rule: NULL

Source: Customer input.

EXTENSION_VALUE

Description: The value for the extended frequency.

Used in: FREQUENCY_DEFN

Type: float

Default: NULL

Rule: NULL

Source: Customer Input.

FREQ_NAME

Description: The name of the sampling frequency.

Used in: FREQUENCY_DEFN

Type: char 30

Default: NULL

Rule: NULL

Source: Customer input.

FREQ_NO

Description: The unique key used within the database to keep the frequencies separate.

Used in: FREQ_VIO_REPORT, FREQUENCY_DEFN, FSS_FREQUENCY

Type: smallint

Default: NULL

Rule: NULL

Source: Code within Ip_InsertFreqDefn.

FSAMPLSPC_NO

Description: The number of this fixed sample space type within this named structure. In the old scheme, still in use in Rev 03.05.01, a fixed sample space is uniquely identified within the tank by the fixed sample space type number and the fixed sample space number, e.g. 1-1, 1-2, 2-1. When adding a new fixed sample space you have to find how many of that type of sample space already exist, then add a new number for the new space.

Used in: RDG_COLCTN_SYS_SNSR_TYPE, RDG_COLCTN_SYS_SENSOR, FIXED_SAMPLE_SPACE, FREQ_VIO_REPORT

Type: smallint

Default: NULL

Rule: NULL

Source: SACS Developers

FSMPLSPCTY_NO

Description: The type of a fixed sample space. This number correclates with the FIXED_SAMPLE_SPACE.FSS_TYPE_NAME column. The owner of this value is the FIXED_SAMPLE_SPACE table.

Used in: RDG_COLCTN_SYS_SNSR, RDG_COLCTN_SYS_SNSR_TYPE
 Alias: FIXED_SAMPLE_SPACE.FSS_TYPE_NO
 Type: tinyint
 Default: NULL
 Rule: NULL
 Source: SACS Developers

FSS_ID

Description: The unique key used within the database to keep the fixed sample spaces unique. The new way of doing things not yet fully implemented in Rev 03.05.01.
 Used in: FSS_FREQUENCY, FSS_SENSOR, FIXED_SAMPLE_SPACE
 Type: int
 Default: NULL
 Rule: NULL
 Source: Generated by database as an identity data type.

FSS_NAME

Description: The name of the fixed sample space. This is the more formal description of the fixed sample space. The description may try to adhere to the Hanford site naming standard, e.g. '241-U-201-RISER-4' or '241-AP-101', or it may be an LOW well number, e.g. '400166'.
 Used in: FIXED_SAMPLE_SPACE
 Type: char 30
 Default: NULL
 Rule: NULL
 Source: Customer configuration information.

FSS_TYPE_NAME

Description: The name of the fixed sample space type.
 Used in: FIXED_SAMPLE_SPACE
 Type: char 30
 Default: NULL
 Rule: NULL
 Values: 'Embedded in Structure' - FSS_TYPE_NO = 4
 'Liquid Observation Well' - 7
 'Riser' - 8
 'Tank' - 1
 'Tank Historical' - 9
 'Catch Tank' - 10

Source: SACS Developers in consultation with the customer.

FSS_TYPE_NO

Description: The type of a fixed sample space. The values for this number are controlled in the FIXED_SAMPLE_SPACE table.
 Used in: FREQ_VIO_REPORT, FIXED_SAMPLE_SPACE
 Alias: FSMPLSPCTY_NO in RDG_COLCTN_SYS_SNSR, RDG_COLCTN_SYS_SNSR_TYPE tables.
 Type: smallint
 Default: NULL
 Rule: NULL
 Values: 1 - Tank
 4 - Embedded in Structure

- 7 - Liquid Observation Well
- 8 - Riser
- 9 - Tank Historical
- 10 - Catch Tank

Source: SACS Developers

GOOD_VALUES

Description: A flag used to say that the referenced reading is near the baseline for that sensor. Near means less than 50% of the way from the baseline to the nearest relative limit.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL

Type: char 1

Default: NULL

Rule: NULL

Values: NULL or not NULL

Source: Calculated by Ip_RetrieveILLDataBatch, Ip_RetrieveSLVLDataBatch, and Ip_RetrieveTPTRDataBatch.

GROUP_END_DTTM

Description: The last day that a tank is a member of a tank group. Tanks are added to groups by inserting a row in the TANK_IN_GROUP table. This row has the start date for the relationship and a NULL end date. When the relationship is terminated a date is inserted in the GROUP_END_DTTM column.

Used in: TANK_IN_GROUP

Type: datetime

Default: NULL

Rule: NULL

Values: NULL or a date-time.

Source: Customer Input.

GROUP_NAME

Description: The name of the database groups to which the related message should be displayed.

Used in: VB_MESSAGE_OTD

Type: char 32

Default: NULL

Rule: NULL

Source: Customer input.

GROUP_START_DTTM

Description: The first day that a tank is a member of a tank group. Tanks are added to groups by inserting a row in the TANK_IN_GROUP table. This row has the start date for the relationship and a NULL end date.

Used in: TANK_IN_GROUP

Type: datetime

Default: NULL

Rule: NULL

HOLDEE

Description: The name of a named structure that is held within another named structure. Usually used for a tank name; the tank is held within the farm.

Used in: NAMED_STRUCTURE_RECUR

Type: char 10

Default: NULL

Rule: NULL

Source: Customer configuration information.

HOLDER

Description: The name of a named structure that holds other named structures. Usually used for a farm name; tanks are held within farms.

Used in: NAMED_STRUCTURE_RECUR

Type: char 10

Default: NULL

Rule: NULL

Source: Customer configuration information.

ILL_CALC_DATE

Description: The date that the ILL value was calculated.

Used in: ILL_VALUE

Type: datetime

Default: NULL

Rule: ill_calc_date_rule

Source: Received from CASS as part of the ILL reading.

ILL_COMMENT

Description: A comment on the ILL value.

Used in: ILL_VALUE

Type: varchar 60

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

ILL_DTTM

Description: The date-time that the ILL scan was taken.

Used in: ILL_VALUE

Type: datetime

Default: NULL

Rule: ill_reading_date_rule

Source: Received from CASS as part of the ILL reading.

ILL_METHOD

Description: An indicator of the method used to calculate the ILL value.

Used in: ILL_VALUE

Type: smallint

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

INCREASE_50

Description: A flag that says that the reading for this row is more than 50% of the way from the baseline to the upper surveillance limit but not lower than the surveillance limit.

Used by the Surveillance Alert Report to select only the rows wanted.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL
 Type: NULL NULL
 Default: NULL
 Rule: NULL
 Values: NULL, not NULL
 Source: Calculated from the BASELINE_CURRENT, REL_UP_LMT, and the reading values.

INCREASE_EX

Description: A flag that says that the reading for this row is above the upper surveillance limit. Used by the Surveillance Alert Report to select only the rows wanted
 Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL
 Type: NULL NULL
 Default: NULL
 Rule: NULL
 Values: NULL, not NULL
 Source: Calculated from the BASELINE_CURRENT, REL_UP_LMT, and the reading values.

INSTALL_DT

Description: The date-time that this sensor was installed in this fixed sample space.
 Used in: FSS_SENSOR
 Type: datetime
 Default: sensor_install_date
 Rule: NULL
 Source: Customer configuration information.

INSTRUMENT

Description: The type of device that was used to take the reading to which a comment was attached. The calculation for this column differs depending on the type of reading.
 Used in: U_COMMENT_REPORT
 Type: char 30
 Default: NULL
 Rule: NULL
 Values: For ILL, copied from the SNSR_TYPE_NAME for the ILL.
 For SLVL, copied from the SNSR_TYPENAME for the surface level reading.
 For TPTR, rtrim(FIXED_SAMPLE_SPACE.COMN_NAME) + ' ' + convert(char(2), RDG_COLCTN_SYS_SENSOR.COLCTN_SNSR_IDX)
 Source: Calculated by Ip_GetRDGComments.

IN_FROM_BOTTOM

Description: The distance from the bottom of the tank to the sensor, in inches. Used mostly with in-tank temperature sensors, it may be null.
 Used in: FSS_SENSOR
 Type: float
 Default: NULL
 Rule: NULL
 Source: Customer configuration information.

LAST_SAMPLE_DTTM

Description: The date-time of the last good sample for this type of sampler for this fixed sample space.

Used in: **FREQ_VIO_REPORT**

Type: datetime

Default: NULL

Rule: NULL

Source: Calculated by Ip_MakeFreqVioRpt and Ip_UserFreqVioRpt.

LIMIT

Description: The number of MEAS_NO units that can pass between readings before some rule is violated. For example, not more than 7 days should pass between temperature readings for a certain tank. LIMIT contains the number 7.

Used in: **FREQUENCY_DEFN**

Type: float

Default: NULL

Rule: NULL

Source: Customer Input.

LIMIT_VIOLATED

Description: A flag used to indicate that a sample-type - fixed-sample-space limit frequency has been violated. Used by the report to select the rows that it wants.

Used in: **FREQ_VIO_REPORT**

Type: char 1

Default: NULL

Rule: NULL

Values: 'Y' - limit has been violated.

' ' - limit has not been violated.

Source: Calculated by Ip_MakeFreqVioRpt.

LOGGING_SEQUENCE_NO

Description: A sequential logging number for each LOW or drywell. The number increments within the LOW.

Used in: **ILL_VALUE**

Type: smallint

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

LOW_WELL_NO

Description: A 6-digit number used to identify the LOW or drywell where a scan was taken.

Used in: **SURV_ALERTS_ILL**

Alias: **ILL_VALUE.LOW_WELL_NUMBER**

Type: NULL NULL

Default: NULL

Rule: NULL

Source: Customer configuration information.

LOW_WELL_NUMBER

Description: A 6-digit number used to identify the LOW or drywell where a scan was taken. The number has 3 distinct sections: xxyyzz, where xx is the tank farm, 10 = A, 11 = AX, 20 = B, 21 = BX, 40 = S, etc.; yy is the number of the tank near which or in which the well is located, 01 = 101, 02 = 102, 14 = 114; and zz is the location of the well in the tank

using a clock face, with 12 pointing north. For wells within tanks, 60 is added to the clock number to distinguish them from drywells. If there is more than one well at a given number, then the second well will have 12 added to the number. For example, well 100160 and 100172 are both in the N side of tank A101 at the 12 o'clock position.

Used in: ILL_VALUE

Type: char 255

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

LWR_LMT_MEAS_CODE

Description: The code for the unit of measure for the lower limit for a reading device when used with a sensor.

Used in: RDGDEV_VALID_SENSOR

Type: char 10

Default: NULL

Rule: NULL

Source: Customer configuration information.

LWR_LMT_MEAS_NO

Description: The number for the unit of measure for the lower limit for a reading device when used with a sensor.

Used in: RDGDEV_VALID_SENSOR

Type: smallint

Default: NULL

Rule: NULL

Source: Customer configuration information.

LWR_LMT_QTY

Description: The value for the lower limit of a reading that a given reading device type can return for a given sensor type.

Used in: RDGDEV_VALID_SENSOR

Type: float

Default: NULL

Rule: NULL

Source: Customer configuration information.

MARKER_METHOD

Description: A code for the marker method used to help establish the ILL value.

Used in: ILL_VALUE

Type: smallint

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

MEASTYP_CODE

Description: The abbreviation for a type of measure. Used to limit the customer choices when setting a unit of measure on a reading or baseline or rate of slope.

Used in: MEASURE

Type: char 10

Default: NULL

Rule: NULL

Values: TPTR - Temperature

DIST - Distance (used with surface level)

TIME - Time

RATE - Rate (used with sloping baselines)

Source: SACS Developers

MEASTYP_NO

Description: The number for the type of measure. Ref. The MEASTYP_CODE.

Used in: MEASURE

Type: tinyint

Default: NULL

Rule: NULL

Values: 1 - TPTR

2 - DIST

3 - TIME

4 - RATE

Source: SACS Developers

MEAS_CODE

Description: A string, usually the commonly accepted abbreviation, that identifies a unit of measure.

Used in: MEAS_BASE_CONVN, MEASURE

Type: char 10

Default: NULL

Rule: NULL

Source: SACS Developers

MEAS_NAME

Description: The name of a unit of measure.

Used in: MEASURE

Type: char 40

Default: NULL

Rule: NULL

Source: SACS Developers

MEAS_NO

Description: A unique number used to keep the units of measure separate.

Used in: TPTR_READING_AUDIT, SLVL_READING_AUDIT, MEAS_BASE_CONVN,

TPTR_READING, SLVL_READING, BASELINES, MEASURE, FREQUENCY_DEFN

Type: smallint

Default: NULL (except ILL_VALUE, which has meas_units_default)

Rule: NULL

Source: SACS Developers

MEAS_TYPE_NO

Description: A number that identifies the type of measure.

Used in: SENSOR

Alias: MEASURE.MEASTYPE_NO

Type: tinyint

Default: NULL

Rule: NULL

Source: Copied from MEASURE.

MESSAGE

Description: The text of the message to display to the users for the message of the day.

Used in: VB_MESSAGE_OTD

Type: char 250

Default: NULL

Rule: NULL

Source: Customer input.

MRDCNAM_NAME

Description: The common name for a manual reading device. This is usually a serial number for a hand-held thermocouple reader.

Used in: MAN_RDG_DEV_COMN_NAME

Type: char 20

Default: NULL

Rule: NULL

Source: Customer input through the temperature input form.

MRDCNAM_NO

Description: The unique number for a manual reading device. Internally maintained.

Used in: MAN_RDG_DEV_COMN_NAME, TPTR_READING_AUDIT, TPTR_READING

Type: smallint

Default: NULL

Rule: NULL

Source: Calculated by Ip_Add_MRDCNAM_NAME.

NO_SLVL_FOUND

Description: A flag that says that a valid surface level could not be found for the 3 months prior to this temperature reading.

Used in: U_TEMPERATURE_REPORTS

Type: char 1

Default: NULL

Rule: NULL

Source: Calculated by Ip_TemperatureReport.

NSTRUCTR_COMN_NAME

Description: The common name for a named structure. This is NOT the Hanford site standard. It is the way that some people prefer to see their tanks named. It is not displayed within SACS.

Used in: NAMED_STRUCTURE

Type: char 20

Default: NULL

Rule: NULL

Values: For a tank it has the structure '101-A','115-SX'.

For a farm it has the structure 'A','U'.

For a catch tank it has the structure '244-U-SUMP','301B-U','X302A-U'.

Source: Customer configuration information.

NSTRUCTR_ID

Description: This is a unique key that defines the named structure. It is NOT the Hanford site standard name for a tank. It is the wya that the Surveillance customer prefers to see their tank names.

Used in: BASELINES, FIXED_SAMPLE_SPACE, FREQ_VIO_REPORT, ILL_VALUE, NAMED_STRUCTURE, NS_COMMENT, NS_COMMENT_AUDIT, RDG_COLCTN_SYS_SENSOR, RDG_COLCTN_SYS_SNSR_TYPE, SLVL_READING, SLVL_READING_AUDIT, SURV_ALERTS_ILL, SURV_ALERTS_SLVL, SURV_ALERTS_TPTR, SURVEILLANCE_ALERTS, TANK_IN_GROUP, TPTR_READING, TPTR_READING_AUDIT

Type: char 10

Default: NULL

Rule: NULL

Values: For a tank it has the structure 'A101','SX115'.

For a farm it has the structure 'A','U'.

For a catch tank it has the structure '244-U-SUMP','U301B','UX302A'.

Source: Customer configuration information

NSTRUCTR_NAME

Description: The is the name of a named structure. This name attempts to follow the Hanford site naming standard for tanks and farms.

Used in: NAMED_STRUCTURE

Type: char 30

Default: NULL

Rule: NULL

Values: For a tank it has the structure '241-A-101', '241-U-101'.

For a farm it has the structure '241-A','241-U'.

For a catch tank it has the structure '244-U-SUMP','241-U-301-B','241-UX-302-A'.

Source: Customer configuration information

NSTRUCTR_TYPE_NAME

Description: The name of the type of named structure. Currently used along with NSTRUCTTYP_NO to differentiate between different kinds of named structures since tank, farms, and catch tanks are all lumped into NAMED_STRUCTURE.

Used in: NAMED_STRUCTURE

Type: char 20

Default: NULL

Rule: NULL

Values: 'Tank' - 1

'Tank Farm' - 2

'Catch Tank' - 3

Source: Customer configuration information.

NSTRUCTTYP_NO

Description: The number for the type of named structure. Currently used along with NSTRUCTTYP_NO to differentiate between different kinds of named structures since tank, farms, and catch tanks are all lumped into NAMED_STRUCTURE.

Used in: RDG_COLCTN_SYS_SNSR_TYPE, RDG_COLCTN_SYS_SENSOR, FIXED_SAMPLE_SPACE, NAMED_STRUCTURE

Type: tinyint

Default: NULL

Rule: NULL

Values: 1 - Tank
 2 - Tank Farm
 3 - Catch Tank
 Source: SACS Developers

NS_COMMENT_DTTM

Description: The date-time to which a named structure comment applies.
 Used in: NS_COMMENT_AUDIT, NS_COMMENT
 Type: datetime
 Default: NULL
 Rule: NULL
 Source: Customer input.

NS_COMMENT_ID

Description: A unique identifier used to tell named structure comments apart.
 Used in: NS_COMMENT_AUDIT, NS_COMMENT
 Type: numeric
 Default: NULL
 Rule: NULL
 Source: Generated by database as an identity data type.

NS_COMMENT_TEXT

Description: The text for the named structure comment.
 Used in: NS_COMMENT_AUDIT, NS_COMMENT
 Type: varchar 250
 Default: NULL
 Rule: NULL
 Source: Customer input.

NS_COM_AUD_ACTION

Description: A code that describes the action that caused a row to be added to the audit trail.
 Used in: NS_COMMENT_AUDIT
 Type: char 1
 Default: NULL
 Rule: NULL
 Values: U - Update. A change was made to the record.
 D - Delete. The record was deleted.
 Source: Triggers for update and delete on NS_COMMENT table.

NS_ENTRY_DTTM

Description: The date-time that a comment was entered into the table. This is distinct from the NS_COMMENT_DTTM.
 Used in: NS_COMMENT_AUDIT, NS_COMMENT
 Type: datetime
 Default: NULL
 Rule: NULL
 Source: Insert trigger on NS_COMMENT table.

OFFSET

Description: When converting one linear unit of measure to another, this is the offset between the zero values. In the conversion from degrees C to F, which is $(C/1.8) + 32$, the OFFSET = 32.

Used in: MEAS_BASE_CONVN

Type: float

Default: NULL

Rule: NULL

Source: SACS Developers.

OUTLIER_READING_COUNT

Description: The number of scan readings outside the expected values.

Used in: ILL_VALUE

Type: smallint

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

PASS_FAIL_REVIEW

Description: The determination by someone on CASS that the LOW or drywell scan is valid.

Used in: ILL_VALUE, SURV_ALERTS_ILL

Type: char 1

Default: NULL

Rule: pass_fail_values_rule

Values: On CASS, the values are 'Y' or 'N'. These are translated when inserted into SACS as 'G' and 'S'.

Source: Received from CASS as part of the ILL reading.

QUALITY_INDICATOR

Description: This is synonymous with the SUSP_DATA_STAT_CODE.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL

Alias: SUSP_DATA_STAT_CODE

Type: char 1

Default: NULL

Rule: NULL

Source: Copied from the readings tables.

RATIO

Description: When converting one linear unit of measure to another, this is the ratio of the slopes between the two units of measure. In the conversion from degrees C to F, which is $(C/1.8) + 32$, the RATIO = 1.8.

Used in: MEAS_BASE_CONVN

Type: float

Default: NULL

Rule: NULL

Source: SACS Developers.

RAW_ILL_VALUE

Description: The uncorrected ILL value.

Used in: ILL_VALUE

Type: float
 Default: NULL
 Rule: NULL
 Source: Received from CASS as part of the ILL reading.

RAW_MARKER_DEPTH

Description: The marker depth without corrections. The marker is a repeatable deviation in the logging that is used to align different loggings.
 Used in: ILL_VALUE
 Type: float
 Default: NULL
 Rule: NULL
 Source: Received from CASS as part of the ILL reading.

RCSYS_ABBREVD_NAME

Description: The abbreviated name for a reading collection system.
 Used in: SURV_ALERTS_TPTR
 Alias: READING_COLCTN_SYSTEM.ABBREVD_NAME
 Type: char 50
 Default: NULL
 Rule: NULL
 Source: Copied from READING_COLCTN_SYSTEM.ABBREVD_NAME.

RCSYS_NAME

Description: The full name of a reading collection system.
 Used in: READING_COLCTN_SYSTEM
 Type: char 50
 Default: NULL
 Rule: NULL
 Source: Customer configuration information.

RCSYS_NO

Description: A unique key used to keep the reading collection systems separate.
 Used in: RDG_COLCTN_SYS_SNSR_TYPE, READING_COLCTN_SYSTEM,
 RDG_COLCTN_SYS_SENSOR, U_SNSR_PTS_RET_PARM, TPTR_READING_AUDIT,
 SLVL_READING_AUDIT, TPTR_READING, SLVL_READING
 Type: smallint
 Default: NULL
 Rule: NULL
 Values: 1 - CASS
 2 - Manually entered data
 3 - HDAT
 4 - TMACS
 Source: SACS Developers.

RDGDEVTYP_NO

Description: A number used to identify a type of reading device.
 Used in: RDG_DEVICE
 Type: smallint
 Default: NULL
 Rule: NULL

Source: SACS Developers.

RDGDEV_NAME

Description: The name of the reading device.

Used in: RDG_DEVICE

Type: char 30

Default: NULL

Rule: NULL

Source: Customer input.

RDGDEV_NO

Description: The unique number for a reading device. This is used inside the database to identify different reading devices.

Used in: MAN_RDG_DEV_COMN_NAME, RDG_DEVICE

Type: smallint

Default: NULL

Rule: NULL

Source: SACS Developers.

RDGDEV_TYPE_NAME

Description: The name of the reading device type. Works in parallel with the RDGDEV_TYP_NO column.

Used in: RDG_DEVICE

Type: char 40

Default: NULL

Rule: NULL

Source: Customer configuration information.

RDG_DEVICE_TYPE_NAME

Description: The name of the reading device type.

Used in: RDGDEV_VALID_SENSOR

Alias: RDGDEV_TYPE_NAME

Type: char 40

Default: NULL

Rule: NULL

Source: Customer configuration information.

RDG_DTTM

Description: The date-time that a reading was taken.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL

Alias: TPTR_DTTM, SLVL_DTTM, ILL_DTTM

Type: datetime

Default: NULL

Rule: NULL

Source: Copied from source tables.

RDG_VALUE

Description: The value for the reading.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL

Alias: TPTR_QTY, SLVL_QTY, CORRECTED_ILL_VALUE

Type: float

Default: NULL

Rule: NULL

Source: Copied from source tables.

REL_LOW_LMT

Description: The lower warning limit for a reading, relative to the baseline established for the reading. The value for this limit will move along with a sloping baseline.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, BASELINES

Type: float

Default: NULL

Rule: rel_low_lmt_rule

Source: Customer input.

REL_UP_LMT

Description: The upper warning limit for a reading, relative to the baseline established for the reading. The value for this limit will move along with a sloping baseline.

Used in: SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, BASELINES

Type: float

Default: NULL

Rule: rel_up_lmt_rule

Source: Customer input.

REMOVAL_DT

Description: The date-time that a sensor was removed from a fixed sample space. Theoretically a sensor may be removed and re-inserted into a sample space many times.

Used in: FSS_SENSOR

Type: datetime

Default: NULL

Rule: NULL

Source: Customer configuration information.

REVIEWED_EMP_ID

Description: The EMP_ID of an employee who reviewed an ILL reading.

Used in: ILL_VALUE

Type: char 7

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

RISER

Description: The name of the riser (a type of fixed sample space) where a temperature reading was taken.

Used in: U_TEMPERATURE_REPORTS

Alias: FIXED_SAMPLE_SPACE.COMN_NAME

Type: char 20

Default: NULL

Rule: NULL

Source: Copied from FIXED_SAMPLE_SPACE.COMN_NAME

RPT_RUN_DTTM

Description: The date-time that the frequency violation report was run.

Used in: **FREQ_VIO_REPORT**

Type: NULL NULL

Default: NULL

Rule: NULL

Source: Calculated by Ip_MakeFreqVioRpt.

RUN_DTTM

Description:

Used in: **SURVEILLANCE_ALERTS, SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, SURV_ALERTS_ILL**

Type: datetime 8

Default: NULL

Rule: NULL

SENSOR_NAME

Description: The name of a type of sensor used to take a reading. Examples are 'Manual FIC' and 'Weight Factor'.

Used in: **FREQ_VIO_REPORT**

Type: char 30

Default: NULL

Rule: NULL

Source: Copied from **SENSOR.SNSR_TYPE_NAME** by Ip_MakeFreqVioRpt.

SLOPE_MEAS_NO

Description: The MEAS_NO for the unit of measure for the slope of a baseline. Refers to **MEASURE.MEAS_NO**.

Used in: **BASELINES**

Alias: **MEASURE.MEAS_NO**

Type: smallint

Default: NULL

Rule: NULL

Source: Customer input.

SLOPE_RATE

Description: The value for the baseline slope rate.

Used in: **BASELINES**

Type: float

Default: NULL

Rule: NULL

Source: Customer input.

SLVL_COMMENT

Description: A comment associated with a single surface-level reading.

Used in: **SLVL_READING_AUDIT, SLVL_READING**

Type: varchar 250

Default: NULL

Rule: NULL

Source: Customer Input.

SLVL_DTIM

Description: The date-time that a surface-level reading was taken.

Used in: SLVL_READING_AUDIT, SLVL_READING

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input.

SLVL_ID

Description: The unique identifier for a surface-level reading.

Used in: SLVL_READING_AUDIT, SLVL_READING

Type: numeric

Default: NULL

Rule: NULL

Source: Generated by database as an identify data type.

SLVL_QTY

Description: The value of the surface-level reading.

Used in: SLVL_READING_AUDIT, SLVL_READING

Type: float

Default: NULL

Rule: NULL

Source: Customer Input.

SNSR_IS_REFERENCE_CODE

Description: A flag to mark the starting sensor in a collection. This sensor usually is numbered 1.

Used in: SENSOR

Type: char 1

Default: NULL

Rule: NULL

Source: Customer configuration information.

SNSR_NO

Description: A unique number used to identify sensors within the database. Only valid within the database.

Used in: RDG_COLCTN_SYS_SENSOR, U_SNSR_PTS_RET_PARM, TPTR_READING_AUDIT, SLVL_READING_AUDIT, S_COMMENT_AUDIT, TPTR_READING, SLVL_READING, FSS_SENSOR, S_COMMENT, SENSOR, BASELINES

Type: int

Default: NULL

Rule: NULL

Source: SACS Developers.

SNSR_RDG_GROUP

Description: The name of the group of sensors to which this type of sensors belongs. For example, there are types of sensors with values 'Auto FIC', 'Manual Tape', and 'Manual FIC'. All of these are members of the 'Surface Level' group when it comes time to set the sample frequency for a tank.

Used in: FREQ_VIO_REPORT, FSS_FREQUENCY, SENSOR_TYPE

Type: char 15

Default: NULL
 Rule: NULL
 Source: SACS Developers.

SNSR_TYPE_CD

Description: A number that identifies the sensor type. Paralled in meaning by SNSR_TYPE_NAME.
 Used in: SENSOR_TYPE
 Type: tinyint
 Default: NULL
 Rule: NULL
 Source: SACS Developers.

SNSR_TYPE_NAME

Description: The name of the sensor type.
 Used in: BASELINES, ILL_VALUE, RDG_COLCTN_SYS_SNSR_TYPE, RDGDEV_VALID_SENSOR, SENSOR, SENSOR_TYPE, SURV_ALERTS_ILL, SURV_ALERTS_SLVL, U_SNSR_PTS_RET_PARM
 Type: char 30
 Default: NULL
 Rule: NULL
 Source: Customer configuration information.

START_DTTM

Description: The starting date-time to display the given message of the day.
 Used in: VB_MESSAGE_OTD
 Type: datetime
 Default: NULL
 Rule: NULL
 Source: Customer input.

START_DTTM

Description: The starting date-time that a baseline is in effect.
 Used in: BASELINES
 Type: datetime
 Default: NULL
 Rule: NULL
 Source: Customer input.

SUSP_DATA_STAT_CODE

Description: A code that indicates whether the reading has been judged to accurately reflect the conditions where it was taken.
 Used in: TPTR_READING_AUDIT, SLVL_READING_AUDIT, FREQ_VIO_REPORT
 Type: char 1
 Default: NULL
 Rule: NULL
 Source: Copied from the readings tables.

SUSP_DATA_STAT_CODE

Description: A code that indicates whether the reading has been judged to accurately reflect the conditions where it was taken.
 Used in: TPTR_READING, SLVL_READING

Type: char 1

Default: rdgs_susp_stat_def

Rule: rdgs_susp_stat_rule

Values: 'T' - transcribed, now evaluation has been made.

'S' - suspect, the value does not reflect the location where it was taken.

'G' - good, the value does reflect the location where it was taken.

Source: Insert triggers set the value to 'S' for values near 0.

Usually set to 'T' on data entry and re-set to 'S' or 'G' manually through the V&V form.

S_COMMENT_DTTM

Description: The date-time to which a sensor comment applies.

Used in: S_COMMENT_AUDIT, S_COMMENT

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input.

S_COMMENT_ID

Description: The unique ID for the sensor comment row.

Used in: S_COMMENT_AUDIT, S_COMMENT

Type: numeric

Default: NULL

Rule: NULL

Source: Generated by database as an identity data type.

S_COMMENT_TEXT

Description: The text of the sensor comment.

Used in: S_COMMENT_AUDIT, S_COMMENT

Type: varchar 255

Default: NULL

Rule: NULL

Source: Customer input.

S_COM_AUD_ACTION

Description: A code that describes the action that caused a row to be added to the audit trail.

Used in: S_COMMENT_AUDIT

Type: char 1

Default: NULL

Rule: NULL

Values: U - Update. A change was made to the record.

D - Delete. The record was deleted.

Source: Triggers for update and delete on the S_COMMENT table.

S_ENTRY_DTTM

Description: The date-time that a comment entry was received by the database.

Used in: S_COMMENT_AUDIT, S_COMMENT

Type: datetime

Default: NULL

Rule: NULL

Source: Insert trigger.

TAG_NAME

Description: The Hanford tag name for a sensor; in a wider sense, how this sensor is known by another data-gathering system. This serves as a translation from data sent by an outside reading collection system to the SNSR_NO used internally.

Used in: TMACS_UNTAGS, RDG_COLCTN_SYS_SENSOR

Type: char 30

Default: NULL

Rule: NULL

Source: Customer configuration information.

TANK

Description: The NSTRUCTR_ID of the tank where the readings were taken.

Used in: U_TEMPERATURE_REPORTS

Alias: NAMED_STRUCTURE.NSTRUCTR_ID

Type: char 10

Default: NULL

Rule: NULL

Source: Copied from NAMED_STRUCTURE.NSTRUCTR_ID

TANK_GROUP_DESC

Description: A text description of a tank group. grouping of tanks.

Used in: TANK_GROUP

Type: text 16

Default: NULL

Rule: NULL

Source: Customer input.

TANK_GROUP_NAME

Description: The name of a group of tanks. This may be an OSR grouping, like 'Flammable Gas' or 'High Heat', or it may be a totally arbitrary grouping of tanks.

Used in: TANK_IN_GROUP, TANK_GROUP

Type: char 30

Default: NULL

Rule: NULL

Source: Customer input.

TEMPERATURE

Description: The TPTR_QTY for a given sensor.

Used in: U_TEMPERATURE_REPORTS

Alias: TPTR_READING.TPTR_QTY

Type: float

Default: NULL

Rule: NULL

Source: Copied from TPTR_READING by Ip_TemperatureReport.

THERMOCOUPLE

Description: The collection index for a temperature sensor.

Used in: U_TEMPERATURE_REPORTS

Type: tinyint

Default: NULL

Rule: NULL

Source: Copied from RDG_COLCTN_SYS_SENSOR.COLCTN_SNSR_IDX by Ip_TemperatureReport.

TOLERANCE

Description: The allowable difference between the corrected ILL value and the baseline before the reading is out of tolerance.

Used in: ILL_VALUE

Type: float

Default: NULL

Rule: NULL

Source: Received from CASS as part of the ILL reading.

TPTR_COMMENT

Description: A comment on a single temperature reading.

Used in: TPTR_READING_AUDIT, TPTR_READING

Type: varchar 250

Default: NULL

Rule: NULL

Source: Customer input.

TPTR_DTTM

Description: The date-time that a temperature reading was taken.

Used in: TPTR_READING_AUDIT, TPTR_READING

Type: datetime

Default: NULL

Rule: NULL

Source: Customer input.

TPTR_ID

Description: The unique ID for a temperature reading.

Used in: TPTR_READING_AUDIT, TPTR_READING

Type: numeric

Default: NULL

Rule: NULL

Source: Generated by database as an identity data type.

TPTR_QTY

Description: The value of the temperature reading.

Used in: TPTR_READING_AUDIT, TPTR_READING

Type: float

Default: NULL

Rule: NULL

Source: Customer input.

TYPE_NAME

Description: The type of thing being measured. Exmaples are: 'Temperature', 'Distance', etc.

Used in: MEASURE

Type: char 40

Default: NULL
Rule: NULL
Source: SACS Developers.

UNITS

Description: The unit of measure used to report a temperature reading.
Used in: U_TEMPERATURE_REPORTS
Alias: MEASURE.MEAS_CODE
Type: char 10
Default: NULL
Rule: NULL
Source: Copied from MEASURE.MEAS_CODE by Ip_TemperatureReport.

UPR_LMT_MEAS_CODE

Description: The unit of measure for the upper limit of reading values that this type of sensor can report.
Used in: RDGDEV_VALID_SENSOR
Alias: MEASURE.MEAS_CODE
Type: char 10
Default: NULL
Rule: NULL
Source: Customer configuration information.

UPR_LMT_MEAS_NO

Description: The internal numeric code for the unit of measure used with the upper limit or reading values that this type of sensor can report.
Used in: RDGDEV_VALID_SENSOR
Alias: MEASURE.MEAS_NO
Type: smallint
Default: NULL
Rule: NULL
Source: Customer configuration information.

UPR_LMT_QTY

Description: The value for the upper limit of readings that this type of reading device can report for this type of sensor.
Used in: RDGDEV_VALID_SENSOR
Type: float
Default: NULL
Rule: NULL
Source: Customer configuration information.

VALUE_IN_TOLERANCE

Description: A flag that shows whether an ILL value is close enough to the baseline or not.
Used in: ILL_VALUE
Type: char 1
Default: NULL
Rule: NULL
Source: Received from CASS as part of the ILL reading.

VB_VERSION_TEXT

Description: A string that contains the version number for the Visual Basic client. This string must match a string within the VB client or the customer cannot use the SACS database. Used to make sure that the revision of the client matches the revision of the database.

Used In: VB_VERSION

Type: char 30

Default: NULL

Rule: NULL

Source: Customer input.

WARNING_VALUE

Description: The number of units of time that can elapse between the last good reading for a fixed sample space and the time that this report was run.

Used In: FREQUENCY_DEFN

Type: float

Default: NULL

Rule: NULL

Source: Customer input.

WARNING_VIOLATED

Description: A flag used to indicate that a sample-type - fixed-sample-space warning frequency has been violated. Used by the report to select the rows that it wants.

Used In: FREQ_VIO_REPORT

Type: char 1

Default: NULL

Rule: NULL

Values: 'Y' - limit has been violated.

' ' - limit has not been violated.

Source: Calculated by lp_MakeFreqVioRpt.

WARN_MEAS_NO

Description: The unit of measure for the warning limit.

Used In: FREQUENCY_DEFN

Alias: MEASURE.MEAS_NO

Type: smallint

Default: NULL

Rule: NULL

Source: Customer input.

3.6 ARCHIVE DATA

No plans for archiving have been made.

4. Project Reference Tables and Appendices

4.0 Hardware and Software Requirements

4.0.1 Server

The server is a Sybase database, version 10.0.2, running on a Sun Sparc 20 computer. The operating system is SunOS 2.2. The memory installed is 64MB with 55 MB allowed for Sybase.

The SACS database is configured for 2.5 GB on — disk drives.

The RAWDATA database is configured for 3.0 GB on — disk drives.

4.0.2 Client

The client runs under MS-Windows 3.1 or Windows for Workgroups 3.11.

Required memory is: 8MB.

Required disk space is: 1 MB for the program and related formatting files. Additional space is required for data retrieval storage.

The Hanford standard ESOE configuration is enough to run the SACS client. The interface is TCP/IP using sockets.

The SACS client can export data to files or, via DDE, to Excel spreadsheets. The cells where the data are sent are described in the SACS SRS document. All of the programming and maintenance of the Excel spreadsheets is done by the customer. It is possible, but not recommended, to run SACS without Excel.

4.1 MS-Windows Dependencies

The setup directories, located \\bcs17\sacspvcs\vbforms\production\{rev_number}, contains the executables and DLLs needed to make the client software work in a MS-Windows environment.

4.2 Visual Basic Constants

4.2.1 CONSTANT.BAS File

This file contains constants used throughout the Visual Basic programming. Only a small portion at the head of the file is copied here.

This file contains:

- APPNAME - the name of the SACS application
- SACS_VERSION - a string that must match the contents of the VB_VERSION row in the database
- INIFILENAME - the name of the application initialization file
- INISTRINGLEN - the length of the initialization strings
- CENTURY - the starting day of this century
- PAST - the earliest date for which the user can manually enter data into SACS

**** Constant.Bas

**** Application Global Constants

```

Global Const APPNAME = "PCSACS"
Global Const SACS_VERSION = "3.5.1"
Global Const INIFILENAME = "PCSACS.INI"
Global Const INISTRINGLEN = 15
Global Const CENTURY = "1/1/1900"
Global Const PAST = "1/1/1950"

```

4.2.2 GLOBAL.BAS File

This file contains the globally-useful information for the Visual Basic client. The contents are:

```

'***
'*** Global.Bas
'***

'*** Handy "Constants"

Global CRLF As String

Global REPORT_PATH As String 'path to the current execution directory
                              'where it is assumed that the Crystal report files also reside

'*****
' These handle the SQL Server connection for this application
'*****

Global DBLIB_VERSION
Global iSqlconn As Integer
Global strServer As String
Global strLoginID As String
Global strPassword As String
Global strLastName As String
Global strDatabaseName As String
Global strProgramName As String
Global bSQLStatus As Integer
Global bNumLockState As Integer '*** Initial State of Num Lock Key when
application started
Global bDBorDBLIBError As Integer '*** SQL Server errors with severity >10 and <=
16
Global bDuplicateIndexError As Integer '*** Duplicate entry message flag
Global iLoginTimeout As Integer
Global iQueryTimeout As Integer
Global strGroup As String '*** Group To Which The User Belongs
Global bPasswordChanged As Integer

Global bConnectionOpen As Integer
Global vTimeLastUsed As Variant
Global vTimeSinceClosed As Variant
Global bPromptForPassword As Integer

```

```
*****
'Global variables for this application
*****
```

```
*** Help Engine
```

```
Global strHelpFilePath As String
```

```
Type MULTIKEYHELP
```

```
mkSize As Integer
```

```
mkKeylist As String * 1
```

```
szKeyphrase As String * 253
```

```
End Type
```

```
*** X/Y Scaling Factors
```

```
Global sXFactor As Single
```

```
Global sYFactor As Single
```

```
*** Windows and System Resources Information
```

```
Global sCPU As String '386, 486, etc.
```

```
Global sMode As String 'Standard or 386 Enhanced
```

```
Global sInitMemFree As String 'Free Memory at Startup
```

```
Global sCoProcessor As String 'Present or Not
```

```
*** Temperature Data Entry Globals
```

```
Global strTempFarm As String 'Input new temp data
```

```
Global iArea_no As Integer 'area number of a given farm
```

```
Global strTempTank As String 'Modify temp data
```

```
Global vTempReadingDate As Variant
```

```
Global vTempReadingTime As Variant
```

```
Global iNumRisers As Integer
```

```
Type TempInRisers
```

```
strNSTRUCTR_ID As String
```

```
strCOMN_NAME As String
```

```
iFSS_TYPE_NO As Integer
```

```
iFSAMPLSPC_NO As Integer
```

```
iCOLCTN_NO As Integer
```

```
iCOLCTN_SNSR_IDX(43) As Integer
```

```
iSNSR_NO(43) As Integer
```

```
iRCSYS_NO(43) As Integer
```

```
strRCSYS_NAME(43) As String
```

```
iTotalSensorsOnRiser As Integer
```

```
iMRDCNAM_NO As Integer
```

```
strComment(43) As String ' Added 5-16-95 Q.
```

```
iPTR_ID(43) As Long ' Added 5-22-95 Q.
```

```
End Type
```

**** Surface Level Entry Globals

Global bChangeExistingSL As Integer
 Global strSLSensorNumbers() As String
 Global strSLRTVSensorNumbers() As String
 Global strSLSensorPoints() As String
 Global strSLData() As String
 Global strErroredSLData() As String

**** V and V Globals

Global strVVTankName As String
 Global strVVStartDate As String
 Global strVVEndDate As String
 Global lTotalVVPoints As Integer
 Global strAllData() As String * 30
 Global strChangedData() As String * 30
 Global strVVSensorPoints() As String
 Global bChangesMade As Integer
 Global strVVReadingType As String
 Global strVVSelectedDataType(2) As String
 Global bVVSlopedBL As Integer ' Bool which indicates if we are working
 with a sloping baseline or not
 Global bVVFlatBL As Integer ' Bool which indicates if we are working
 with a level baseline or not
 Global sVVBLSlope As Single ' Value of rate of change of slope
 Global sVVBLLUpLim As Single ' limit value above baseline; both sloped
 and level
 Global sVVBLLoLim As Single ' limit value below baseline; both sloped
 and level
 Global sVVBValue As Single ' value at the beginning of the baseline
 Global strVVBDate As String ' date at which a baseline begins
 Global strVVBLSlopeDate As String ' date at which a sloping baseline
 begins its slope

**** ' # of days from the start of the baseline to the start of the query period
 or the start of the X axis resize

**** if axis has been resized. + value means baseline started before
 beginning of display period.

**** - value means baseline started after beginning of
 display period

Global lVVBDaysFromStart As Integer

**** ' # of days from the start of the baseline to the end of the query period
 or the end of the X axis resize

**** if axis has been resized. + value means baseline started before
 ending of display period.

**** - value means baseline started after end of display
 period.

Global IVVBLDaysFromEnd As Integer
 Global IVVSlopedBLDaysFromStart As Integer
 Global IVVSlopedBLDaysFromEnd As Integer
 Global bVVBLEnabled As Integer ' true-baselines are enabled if only 1
 sensor was retrieved.

' false-no baselines in either plot or grid if more than
 one sensor was retrieved.

Global strVVBLSensor As String

	' Temperature	Surface Levels	ILL
Global IVVGrdCFG1 As Integer	' Riser #	UNUSED	LOW
WELL			
Global IVVGrdCFG2 As Integer	' TC #	Collection	Seq #
Global IVVGrdDate As Integer	' Date Column		
Global IVVGrdBLValue As Integer	' Baseline value column		
Global IVVGrdBLUp As Integer	' Baseline Value + Upper Limit Column		
Global IVVGrdValue As Integer	' Value Column		
Global IVVGrdBLLo As Integer	' Baseline Value - Lower Limit Column		
Global IVVGrdQI As Integer	' Quality Indicator Column		
Global IVVGrdReviewedBy As Integer	' Reviewed By Column		
Global IVVGrdOtherComment As Integer	' Other Comments Column		
Global IVVGrdRdgComment As Integer	' Reading Comment Column		
Global IVVGrdIndex As Integer	' Index from Grid into Data Arrays		
Column			
Global IVVGrdNumCols As Integer	' the number of columns in the grid		
(varies on baselines or no)			
Global IVVAlldataGS As Integer			
Global IVVGraphTop As Integer			
Global IVVGraphLeft As Integer			
Global IVVGraphWidth As Integer			
Global IVVGraphHeight As Integer			
Global bQueryBLEnabled As Integer	' If true the user successfully retrieved		
a BL from the frmRetrieveBL form			

'new and improved V and V

Global dAmplitude() As Double

Global dDistance() As Double

Global iAux() As Integer

Global iGSColor() As Integer

Global iGSSymbol() As Integer

Global strXLabels As String

Global dMaxAmp As Double

Global dMinAmp As Double

Global dMinDist As Double

Global dMaxDist As Double

Global dXstart As Double

Global dXend As Double

Global dXTickSpacing As Double

Global strTypeXTicks As String

Global dYTickSpacing As Double

Global bXGridLines As Integer

```

Global bYGridLines As Integer
Global dOriginalMaxAmp As Double
Global dOriginalMinAmp As Double
Global dOriginalMinDist As Double
Global dOriginalMaxDist As Double

```

```

****

```

```

**** Query Globals

```

```

****

```

```

Global Const FILE_TEMP_DATA = "~tmp.dat"
Global Const COMMENT_FILE = "C:\WPDATA\COMMENTS.TXT"
Global iOptionFrequency As Integer
Global strTankName As String
Global strStartDate As String
Global strEndDate As String
Global iTotalPoints As Integer
Global strSensorPoints() As String
Global strQBLValue As String
Global strQBDate As String
Global strQBSlope As String
Global strQBLUpper As String
Global strQBLLower As String
Global strQueryBLDataType As String

```

```

****

```

```

**** Reports

```

```

****

```

```

Global strReportType As String
Global strFontName As String
Global bReportsUsedThisSession As Integer
Global Const FILE_BASELINE = "~bltmp.dat"
Global Const FILE_FREQUENCY = "~freqrpt.dat"

```

```

****

```

```

**** User Defined Data Types

```

```

****

```

```

**** User defined data type for SqlGetColumnInfo
Type ColumnData
    Coltype As Integer
    ColLen As Long
    Colname As String * 30
    ColSqlType As String * 30
End Type

```

```

**** User defined data type for SqlGetAltColInfo
Type AltColumnData
    ColID As Integer

```

```

DataType As Integer
MaxLen As Long
AggType As Integer
AggOpName As String ^ 30
End Type

```

```

*** User defined data type for SqlBcpColumnFormat

```

```

Type BcpColData
  FType As Integer
  FPLen As Integer
  fColLen As Long
  FTerm As String * 30
  FTLen As Integer
  TCol As Integer
End Type

```

```

*** User defined data type for SqlDateCrack

```

```

Type DateInfo
  Year As Integer
  Quarter As Integer
  Month As Integer
  DayOfYear As Integer
  Day As Integer
  Week As Integer
  WeekDay As Integer
  Hour As Integer
  Minute As Integer
  Second As Integer
  Millisecond As Integer
End Type

```

```

*****
These are globals that apply to the starting of
the comment maintenance form.
*****

```

```

Global lCalledForm As Integer
Global strCommentTankName As String
Global strCommentSensorName As String
Global strCommentDate As String
Global strCommentTime As String

```

```

Global strSensorInformation() As String

```

```

Global bOptionClickCanceled As Integer

```

```

*****
These are globals that are used to pass the
report date from frmFrequency to frmFrequencyViolation
and date from frmBaselineReport to frmSurveillanceAlert

```

Global strRptDate As String

These are general globals that are used in lots of places

Global strTemp As String

Global lTemp As Integer

Global iRV As Integer

4.3 Visual Basic External Products - VBX files

Additional products that are used within Visual Basic are:

- Chart Builder - from Pinnacle Publishing. Used for displaying graphs on-screen (V&V form).
- Crystal Reports - from Seagate Software. Used for report formatting and some data selection (Max Temperature Report).
- Graphics Server - from Pinnacle Publishing. Used for displaying graphic information on screen.

4.4 Visual Basic Forms call Visual Basic Forms

Table 5. Visual Basic forms that call other Forms

This table lists the calling file and the form that is called sorted into order by the calling file.

Calling File	Called Form	# of times called
blreport.frm	frmSurveillanceAlert	2
config.frm	frmGroupDesc	1
freq.frm	frmFrequencyViolation	1
freqvio.frm	frmSelFont	1
login.frm	mdlMain	1
main.frm	frmEnterPWD	1
main.frm	frmLogin	1
mdlmain.frm	frmAbout	1
mdlmain.frm	frmBaseline	1
mdlmain.frm	frmBaselineReport	1
mdlmain.frm	frmCommentInputModification	1
mdlmain.frm	frmCommentReport	1
mdlmain.frm	frmConfig	1
mdlmain.frm	frmFrequency	1
mdlmain.frm	frmMaxTempRpt	1
mdlmain.frm	frmMessages	1
mdlmain.frm	frmPassword	1
mdlmain.frm	frmQuery	1
mdlmain.frm	frmReportSelection	1
mdlmain.frm	frmSampleFreq	1
mdlmain.frm	frmSLVLRpt	1
mdlmain.frm	frmSurfaceLevel	1
mdlmain.frm	frmTempInput	1

mdlmain.frm	frmTempModify	1
mdlmain.frm	frmTempReport	1
mdlmain.frm	frmTempSelection	1
mdlmain.frm	frmVVget	1
mdlmain.frm	mdlMain	1
query.frm	frmCommentReport	1
query.frm	frmRetrieveBL	1
sampfreq.frm	frmFreqDefn	1
slinput.frm	frmCommentInputModification	1
slinput.frm	frmReadingComment	1
slinput.frm	frmSlvIget	1
splash.frm	frmLogin	1
suralert.frm	frmSelfFont	1
tempcpt.frm	frmSelfFont	1
tpinput.frm	frmCommentInputModification	1
tpinput.frm	frmReadingComment	1
tpmodify.frm	frmCommentInputModification	1
tpmodify.frm	frmReadingComment	1
vvget.frm	frmVVMMain	1
vvgraph.frm	frmVVMMain	1
vvgrid.frm	frmAllDataGS	1
vvgrid.frm	frmOtherComments	1
vvgrid.frm	frmVVMMain	1
vvmain.frm	frmVVGraph	1
vvmain.frm	frmVVgrid	1
vvmain.frm	frmVVScale	1

Table 6. Visual Basic forms called by other Forms

This table lists the Visual Basic forms that are called by other forms, sorted by the form that is called.

Called Form	Calling File	# of times called
frmAbout	mdlmain.frm	1
frmAllDataGS	vvgrid.frm	1
frmBaseline	mdlmain.frm	1
frmBaselineReport	mdlmain.frm	1
frmCommentInputModification	mdlmain.frm	1
frmCommentInputModification	slinput.frm	1
frmCommentInputModification	tpinput.frm	1
frmCommentInputModification	tpmodify.frm	1
frmCommentReport	mdlmain.frm	1
frmCommentReport	query.frm	1
frmConfig	mdlmain.frm	1
frmEnterPWD	main.frm	1
frmFreqDefn	sampfreq.frm	1
frmFrequency	mdlmain.frm	1
frmFrequencyViolation	freq.frm	1
frmGroupDesc	config.frm	1
frmLogin	main.frm	1

frmLogin	splash.frm	1
frmMaxTempRpt	mdlmain.frm	1
frmMessages	mdlmain.frm	1
frmOtherComments	vvgrid.frm	1
frmPassword	mdlmain.frm	1
frmQuery	mdlmain.frm	1
frmReadingComment	slinput.frm	1
frmReadingComment	tpinput.frm	1
frmReadingComment	tpmodify.frm	1
frmReportSelection	mdlmain.frm	1
frmRetrieveBL	query.frm	1
frmSampleFreq	mdlmain.frm	1
frmSelFont	freqvio.frm	1
frmSelFont	suralert.frm	1
frmSelFont	temprrpt.frm	1
frmSlvlget	slinput.frm	1
frmSLVLRpt	mdlmain.frm	1
frmSurfaceLevel	mdlmain.frm	1
frmSurveillanceAlert	blreport.frm	2
frmTempInput	mdlmain.frm	1
frmTempModify	mdlmain.frm	1
frmTempReport	mdlmain.frm	1
frmTempSelection	mdlmain.frm	1
frmVVget	mdlmain.frm	1
frmVVGraph	vvmain.frm	1
frmVVgrid	vvmain.frm	1
frmVVMaIn	vvget.frm	1
frmVVMaIn	vvgraph.frm	1
frmVVMaIn	vvgrid.frm	1
frmVVSale	vvmain.frm	1
mdlMain	login.frm	1
mdlMain	mdlmain.frm	1

4.5 Visual Basic Forms call Stored Procedures

Sorted by form name and stored procedure name.

Table 7. Visual Basic forms call Stored Procedures

VB Form File	Stored Procedure Called	# of times in form
baseline.frm	lp_DeleteBaseline	1
baseline.frm	lp_GetAllSensorPts	1
baseline.frm	lp_GetAllSensorTypes	1
baseline.frm	lp_GetAllTankNames	1
baseline.frm	lp_GetSensorGroupBySensorType	1
baseline.frm	lp_InsertBaseline	3
baseline.frm	lp_RetrieveVVBaselines	2
baseline.frm	lp_UpdateBaseline	2
blreport.frm	lp_DisSurvAlertsBatchRptILL	1
blreport.frm	lp_DisSurvAlertsBatchRptSLVL	1
blreport.frm	lp_DisSurvAlertsBatchRptTPTR	1

VB Form File	Stored Procedure Called	# of times in form
blreport.frm	lp_GetAllTankNames	1
blreport.frm	lp_GetBatchTime	1
blreport.frm	lp_GetSnsrRdgGroup	1
blreport.frm	lp_GetSurvAlertsRptILL	1
blreport.frm	lp_GetSurvAlertsRptSLVL	1
blreport.frm	lp_GetSurvAlertsRptTPTR	1
blretrieve.frm	lp_GetSensorGroupBySensorType	1
blretrieve.frm	lp_GetSensorGroupBySensorType	1
blretrieve.frm	lp_GetSensorNumber	1
blretrieve.frm	lp_RetrieveVVBaselines	2
blretrieve.frm	lp_RetrieveVVBaselines	1
cmntmod.frm	lp_DeleteNamedStructureComment	1
cmntmod.frm	lp_DeleteSensorComment	1
cmntmod.frm	lp_GetAllSensorPtsCommentMaint	1
cmntmod.frm	lp_GetAllSensorTypes	1
cmntmod.frm	lp_GetAllTankNames	1
cmntmod.frm	lp_GetNamedStructureComment	1
cmntmod.frm	lp_GetSensorComment	1
cmntmod.frm	lp_InsertNamedStructureComment	1
cmntmod.frm	lp_InsertSensorComment	1
cmntmod.frm	lp_ModifyNamedStructureComment	1
cmntmod.frm	lp_ModifySensorComment	1
cmnttrpt.frm	lp_DeleteCommentReportTable	1
cmnttrpt.frm	lp_GetAllSensorPts	1
cmnttrpt.frm	lp_GetAllSensorTypes	1
cmnttrpt.frm	lp_GetAllTankNames	1
cmnttrpt.frm	lp_GetNSTRUCTRComments	1
cmnttrpt.frm	lp_GetRDGComments	1
cmnttrpt.frm	lp_GetSNSRComments	1
config.frm	lp_AddTankToGroup	1
config.frm	lp_GetAllTankGroups	1
config.frm	lp_GetAllTankGroups	1
config.frm	lp_GetAllTankNames	1
config.frm	lp_GetTanksInGroup	1
config.frm	lp_RemoveTankFromGroup	1
freq.frm	lp_GetFreqVioRptDate	1
freq.frm	lp_GetFreqVioRptFlt	1
freq.frm	lp_GetFreqVioSnsrGrps	1
freq.frm	lp_MakeFreqVioRpt	1
freq.frm	lp_UserFreqVioRpt	1
freqdefn.frm	lp_DelFreqDefn	1
freqdefn.frm	lp_GetAllFreqDefnNames	1
freqdefn.frm	lp_GetAllFreqDefnNames	2
freqdefn.frm	lp_GetFreqDefn	1
freqdefn.frm	lp_GetTimeMeasCode	1
freqdefn.frm	lp_InsertFreqDefn	1
groupdes.frm	lp_DeleteTankGroup	1
groupdes.frm	lp_GetAllTankGroups	1
groupdes.frm	lp_GetAllTankGroups	2
groupdes.frm	lp_GetTankGroupDesc	1

VB Form File	Stored Procedure Called	# of times in form
groupdes.frm	lp_GetTanksInGroup	1
main.frm	lp_Add_MRDCNAM_NAME	1
main.frm	lp_Is_MRDCNAM_NAME_Valid	1
main.frm	lp_LockRpt	1
main.frm	lp_UnlockRpt	1
mdlmain.frm	lp_GetMessages	2
mdlmain.frm	lp_GetVersion	1
message.frm	lp_AddMessage	1
othrcmnt.frm	lp_GetTankAndSensorComment	1
query.frm	lp_GetAllTankNames	1
query.frm	lp_GetSensorPts	1
query.frm	lp_GetSensorTypes_GST	1
query.frm	lp_RetrieveData	1
query.frm	lp_RetrieveTPTR_byTC	1
sampfreq.frm	lp_DeIFSSSFRow	1
sampfreq.frm	lp_GetAllFreqDefnNames	1
sampfreq.frm	lp_GetAllFreqDefnNames	1
sampfreq.frm	lp_GetAllTankNames	1
sampfreq.frm	lp_GetFSSSFTank	1
sampfreq.frm	lp_InsertFSSSFRow	1
slinput.frm	lp_Delete_SLVL_RDG	1
slinput.frm	lp_GetAllTankNames	1
slinput.frm	lp_GetManualSensorTypes_SLVL	1
slinput.frm	lp_InsertSLVL	1
slinput.frm	lp_UpdateSLVL	2
slrtv.frm	lp_GetAllTankNames	1
slrtv.frm	lp_GetSensorPts	1
slrtv.frm	lp_GetSensorTypes_SLVL	1
slrtv.frm	lp_RetrieveSLVLData	1
sivlrpt.frm	lp_GetTankName	1
temp_rpt.frm	lp_GetAllTankGroups	1
temp_rpt.frm	lp_TemperatureReport	1
temprrpt.frm	lp_Get_COLCTN_NO	1
temprrpt.frm	lp_Get_Num_TPTR_Sensors	1
temprrpt.frm	lp_Get_TPTR_RDGS	1
temprrpt.frm	lp_GetCurrentGoodSLVL	1
temprrpt.frm	lp_GetSnsrSpacing	1
temprrpt.frm	lp_GetTankName	1
tpfarm.frm	lp_get_farm	1
tpfarm.frm	lp_Get_Num_TPTR_Risers	1
tpfarm.frm	lp_Get_TPTR_Time	1
tpfarm.frm	lp_GetTankName	1
tpinput.frm	lp_Get_RDG_DEVICE_NAME	3
tpinput.frm	lp_Get_TPTR_Risers	1
tpinput.frm	lp_Get_TPTR_Sensors	1
tpinput.frm	lp_InsertTPTR	1
tpmodify.frm	lp_Delete_TPTR_RDG	1
tpmodify.frm	lp_Get_MRDCNAM_NAME_byTank	1
tpmodify.frm	lp_Get_Num_TPTR_Risers_byTank	1
tpmodify.frm	lp_Get_TPTR_Data	1

VB Form File	Stored Procedure Called	# of times in form
tpmodify.frm	lp_Get_TPTR_Risers_byTank	1
tpmodify.frm	lp_Get_TPTR_Sensors_byTank	1
tpmodify.frm	lp_UpdateTPTR	1
vvget.frm	lp_Drop_User_SnsrPtsRetParm	1
vvget.frm	lp_Get_User_SnsrPtsRetParm	1
vvget.frm	lp_GetAllTankNames	1
vvget.frm	lp_GetSensorPts	1
vvget.frm	lp_GetSensorTypes	1
vvget.frm	lp_GetTankAndSensorCommentCnt	1
vvget.frm	lp_Load_User_SnsrPtsRetParm	1
vvmain.frm	lp_Load_VVReadBatchUpdt	1
vvmain.frm	lp_RetrieveVVBaselines	1
vvmain.frm	lp_Updt_SuspDataStatCode	1

4.6 Stored Procedures called by Visual Basic Forms

Sorted by stored procedure name and form name.

Table 8. Stored Procedures called by Visual Basic Forms

Stored Procedure Called	VB Form File	# of times in form
lp_Add_MRDCNAM_NAME	main.frm	1
lp_AddMessage	message.frm	1
lp_AddTankToGroup	config.frm	1
lp_Delete_SLVL_RDG	slinput.frm	1
lp_Delete_TPTR_RDG	tpmodify.frm	1
lp_DeleteBaseline	baseline.frm	1
lp_DeleteCommentReportTable	cmnt rpt.frm	1
lp_DeleteNamedStructureComment	cmntmod.frm	1
lp_DeleteSensorComment	cmntmod.frm	1
lp_DeleteTankGroup	groupdes.frm	1
lp_DelFreqDefn	freqdefn.frm	1
lp_DelFSSSFRow	sampfreq.frm	1
lp_DisSurvAlertsBatchRptILL	blreport.frm	1
lp_DisSurvAlertsBatchRptSLVL	blreport.frm	1
lp_DisSurvAlertsBatchRptTPTR	blreport.frm	1
lp_Drop_User_SnsrPtsRetParm	vvget.frm	1
lp_Get_COLCTN_NO	temprpt.frm	1
lp_get_farm	tpfarm.frm	1
lp_Get_MRDCNAM_NAME_byTank	tpmodify.frm	1
lp_Get_Num_TPTR_Risers	tpfarm.frm	1
lp_Get_Num_TPTR_Risers_byTank	tpmodify.frm	1
lp_Get_Num_TPTR_Sensors	temprpt.frm	1
lp_Get_RDG_DEVICE_NAME	tpinput.frm	3
lp_Get_TPTR_Data	tpmodify.frm	1
lp_Get_TPTR_RDGS	temprpt.frm	1
lp_Get_TPTR_Risers	tpinput.frm	1
lp_Get_TPTR_Risers_byTank	tpmodify.frm	1
lp_Get_TPTR_Sensors	tpinput.frm	1

Stored Procedure Called	VB Form File	# of times in form
lp_Get_TPTR_Sensors_byTank	tpmodify.frm	1
lp_Get_TPTR_Time	tpfarm.frm	1
lp_Get_User_SnsrPtsRetParm	vvget.frm	1
lp_GetAllFreqDefnNames	freqdefn.frm	1
lp_GetAllFreqDefnNames	freqdefn.frm	2
lp_GetAllFreqDefnNames	sampfreq.frm	1
lp_GetAllFreqDefnNames	sampfreq.frm	1
lp_GetAllSensorPts	baseline.frm	1
lp_GetAllSensorPts	cmntrpt.frm	1
lp_GetAllSensorPtsCommentMaint	cmntmod.frm	1
lp_GetAllSensorTypes	baseline.frm	1
lp_GetAllSensorTypes	cmntmod.frm	1
lp_GetAllSensorTypes	cmntrpt.frm	1
lp_GetAllTankGroups	config.frm	1
lp_GetAllTankGroups	config.frm	1
lp_GetAllTankGroups	groupdes.frm	1
lp_GetAllTankGroups	groupdes.frm	2
lp_GetAllTankGroups	temp_rpt.frm	1
lp_GetAllTankNames	baseline.frm	1
lp_GetAllTankNames	blreport.frm	1
lp_GetAllTankNames	cmntmod.frm	1
lp_GetAllTankNames	cmntrpt.frm	1
lp_GetAllTankNames	config.frm	1
lp_GetAllTankNames	query.frm	1
lp_GetAllTankNames	sampfreq.frm	1
lp_GetAllTankNames	slinput.frm	1
lp_GetAllTankNames	slrtv.frm	1
lp_GetAllTankNames	vvget.frm	1
lp_GetBatchTime	blreport.frm	1
lp_GetCurrentGoodSLVL	temp_rpt.frm	1
lp_GetFreqDefn	freqdefn.frm	1
lp_GetFreqVloRptDate	freq.frm	1
lp_GetFreqVloRptFlt	freq.frm	1
lp_GetFreqVloSnsrGrps	freq.frm	1
lp_GetFSSSTank	sampfreq.frm	1
lp_GetManualSensorTypes_SLVL	slinput.frm	1
lp_GetMessages	mdlmain.frm	2
lp_GetNamedStructureComment	cmntmod.frm	1
lp_GetNSTRUCTRComments	cmntrpt.frm	1
lp_GetRDGComments	cmntrpt.frm	1
lp_GetSensorComment	cmntmod.frm	1
lp_GetSensorGroupBySensorType	baseline.frm	1
lp_GetSensorGroupBySensorType	blretve.frm	1
lp_GetSensorGroupBySensorType	blretve.frm	1
lp_GetSensorNumber	blretve.frm	1
lp_GetSensorPts	query.frm	1
lp_GetSensorPts	slrtv.frm	1
lp_GetSensorPts	vvget.frm	1
lp_GetSensorTypes	vvget.frm	1
lp_GetSensorTypes_GST	query.frm	1

Stored Procedure Called	VB Form File	# of times in form
lp_GetSensorTypes_SLVL	slrtv.frm	1
lp_GetSNSRComments	cmntprt.frm	1
lp_GetSnsRdgGroup	blreport.frm	1
lp_GetSnsSpacing	tempprt.frm	1
lp_GetSurvAlertsRptILL	blreport.frm	1
lp_GetSurvAlertsRptSLVL	blreport.frm	1
lp_GetSurvAlertsRptTPTR	blreport.frm	1
lp_GetTankAndSensorComment	othrcmnt.frm	1
lp_GetTankAndSensorCommentCnt	vvget.frm	1
lp_GetTankGroupDesc	groupdes.frm	1
lp_GetTankName	slvprt.frm	1
lp_GetTankName	tempprt.frm	1
lp_GetTankName	tpfarm.frm	1
lp_GetTanksInGroup	config.frm	1
lp_GetTanksInGroup	groupdes.frm	1
lp_GetTimeMeasCode	freqdefn.frm	1
lp_GetVersion	mdlmain.frm	1
lp_InsertBaseline	baseline.frm	3
lp_InsertFreqDefn	freqdefn.frm	1
lp_InsertFSSSFRow	sampfreq.frm	1
lp_InsertNamedStructureComment	cmntmod.frm	1
lp_InsertSensorComment	cmntmod.frm	1
lp_InsertSLVL	slinput.frm	1
lp_InsertTPTR	tpinput.frm	1
lp_Is_MRDCNAM_NAME_Valid	main.frm	1
lp_Load_User_SnsRptsRetParm	vvget.frm	1
lp_Load_VVReadBatchUpdt	vvmain.frm	1
lp_LockRpt	main.frm	1
lp_MakeFreqVioRpt	freq.frm	1
lp_ModifyNamedStructureComment	cmntmod.frm	1
lp_ModifySensorComment	cmntmod.frm	1
lp_RemoveTankFromGroup	config.frm	1
lp_RetrieveData	query.frm	1
lp_RetrieveSLVLData	slrtv.frm	1
lp_RetrieveTPTR_byTC	query.frm	1
lp_RetrieveVVBaselines	baseline.frm	2
lp_RetrieveVVBaselines	blrtv.frm	2
lp_RetrieveVVBaselines	blrtv.frm	1
lp_RetrieveVVBaselines	vvmain.frm	1
lp_TemperatureReport	temp_rpt.frm	1
lp_UnlockRpt	main.frm	1
lp_UpdateBaseline	baseline.frm	2
lp_UpdateSLVL	slinput.frm	2
lp_UpdateTPTR	tpmodify.frm	1
lp_Updt_SuspDataStatCode	vvmain.frm	1
lp_UserFreqVioRpt	freq.frm	1

4.7 Stored Procedures called by Visual Basic Forms

Sorted by stored procedure name and form name.

Table 9. Stored Procedures called by UNIX Scripts

Stored Procedure Called	Calling UNIX Script
Ip_Add_TMACS_rdg	DoRDBMS.sh
Ip_GetSurvAlertsBatchRpt	ProcessSurfaceLevels.sh
Ip_MakeFreqVioRpt	ProcessSurfaceLevels.sh

4.8 Sybase Internal Dependencies

The Sybase internal dependencies are not listed here for the stored procedures and tables. The current dependencies can be retrieved by using the *sp_depends* command within Sybase. This command will show what stored procedures and triggers reference a given table or stored procedure. Refer to the Sybase manuals for more information.

4.9 Database Stored Procedures

4.9.1 Ip_AddMessage - Stored Procedure

- Purpose

Adds a new row to the VB_MESSAGE_OTD table.

- Function

Same.

- Subordinates

None.

- Dependencies

Uses the VB_MESSAGE_OTD table.

- Interface

Input parameters:

@group (char 32) The database user group that will receive the message.

@message (char 250) The text of the message.

@start_dttm (datetime) The date-time to start displaying the message.

@end_dttm (datetime) The date-time to stop displaying the message.

@emp_id (char 7) The ID of the employee who wrote the message.

Returned values:

None.

- Resources

- Processing

- Data

4.9.2 Ip_AddTankToGroup - Stored Procedure

- Purpose

Add a tank to a tank group.

- Function

Same.

- Subordinates

None.

○ Dependencies

○ Interface

Input parameters:

@group_name (char 30) The name of the tank group to which the tank is added.

@tank_name (char 10) The name (NSTRUCTR_ID) of the tank to add to the group.

@start_dttm (datetime) The date and time that the tank was added. If NULL, the proc inserts the current system date and time.

Returned values:

None.

- Resources
- Processing
- Data

4.9.3 Ip_Add_MRDCNAM_NAME - Stored Procedure

○ Purpose

This procedure adds a new row to the MAN_RDG_DEV_COMN_NAME table and returns the value of the new MRDCNAM_NO.

○ Function

Inserts a row into MAN_RDG_DEV_COMN_NAME table. Selects the maximum value for MRDCNAM_NO and returns it.

○ Subordinates

None.

○ Dependencies

Inserts into, selects from MAN_RDG_DEV_COMN_NAME.

○ Interface

Input parameters:

@mrdcnam_name (char 20) The common name for a manual reading device.

Returned values:

Maximum value of the MRDCNAM_NO in the MAN_RDG_DEV_COMN_NAME table.

- Resources
- Processing
- Data

4.9.4 Ip_Add_TMACS_rdg - Stored Procedure

○ Purpose

Stores data in SACS that was gathered by TMACS.

○ Function

The TMACS data file is reformatted and the lines are passed one at a time to this stored procedure. The procedure then:

Converts the reading system name to a SACS number, e.g. TMACS = 4.

Uses the tag name to extract the SACS sensor number.

Uses the TMACS sampler status code to set the SUSP_DATA_STAT_CODE for SACS.

"GOOD" -> T, "UNKNOWN" -> S.

Converts the name of the measure to the SACS internal measure number.

Inserts the reading in the appropriate table - TPTR_READING or SLVL_READING.

○ Subordinates

None.

- ~ Dependencies

- Selects from:

- MEASURE
 - RDG_COLCTN_SYS_SENSOR
 - READING_COLCTN_SYSTEM
 - SENSOR
 - SENSOR_TYPE

- Inserts into:

- SLVL_READING
 - TPTR_READING

- Interface

- Input parameters:

- @RDG_SYS_NM (char 10) The reading system that took the reading. In this case, "TMACS" as a string.
 - @RDG_DTTM (datetime) The date and time that the reading was taken.
 - @TAG_NAME (char 30) The tag name for the reading.
 - @RDG_QTY (float 8) The value for the reading.
 - @MEAS_NM (char 10) The name of the unit of measure for the reading.
 - @SUSP_DATA_STAT_CODE (char 20) The sampler status code. For TMACS, may be "GOOD" or "UNKNOWN".

- Returned values:

- None.

- Resources
- ~ Processing
- ~ Data

4.9.5 Ip_DeIFSSSFRow - Stored Procedure

- Purpose

- Deletes a row in the FSS_.

- Function

- Subordinates

- None.

- ~ Dependencies

- ~ Interface

- Input parameters:

- @tank_name (char 10)
 - @fss_comn_name (char 20)
 - @snsr_rdg_group (char 15)

- Returned values:

- ~ Resources
- Processing
- ~ Data

4.9.6 Ip_DeIFreqDefn - Stored Procedure

- Purpose

- Deletes a frequency definition.

- Function

Receives the name of a frequency definition and deletes that row(s) from the table.

- Subordinates

None.

- Dependencies

Deletes from FREQUENCY_DEFN.

- Interface

Input parameters:

@freq_name (char 30) The name of the frequency definition to delete.

Returned values:

None.

- Resources
- Processing
- Data

4.9.7 Ip_DeleteBaseline - Stored Procedure

- Purpose

Deletes a baseline for a tank and type of sensor.

- Function

Same.

- Subordinates

None.

- Dependencies

Delete on BASELINES.

- Interface

Input parameters:

@TankName (char 10) The name (NSTRUCTR_ID) of the tank.

@SnsrTypeName (char 30) The type of sensor.

@SnsrNo (Int) The sensor number.

Returned values:

None.

- Resources
- Processing
- Data

4.9.8 Ip_DeleteCommentReport - Stored Procedure

- Purpose

Deletes the contents of the U_COMMENT_REPORT table.

- Function

Same.

- Subordinates

None.

- Dependencies

Delete on U_COMMENT_REPORT.

- Interface

Input parameters:

None.

Returned values:

None.

- Resources
- Processing
- Data

4.9.9 Ip_DeleteNamedStructureComment - Stored Procedure

~ Purpose

Deletes a comment from the NS_COMMENT table.

~ Function

Same.

~ Subordinates

None.

~ Dependencies

Deletes from NS_COMMENT table.

~ Interface

Input parameters:

@Comment_ID (numeric) Identity key of the comment to delete.

Returned values:

None.

• Resources

~ Processing

~ Data

4.9.10 Ip_DeleteSensorComment - Stored Procedure

~ Purpose

Deletes a comment from the S_COMMENT table.

~ Function

Same.

~ Subordinates

None.

~ Dependencies

Deletes from S_COMMENT table.

• Interface

Input parameters:

@Comment_ID (numeric) Identity key of the comment to delete.

Returned values:

None.

~ Resources

• Processing

~ Data

4.9.11 Ip_DeleteTankGroup - Stored Procedure

~ Purpose

Deletes a row from the TANK_GROUP table.

~ Function

Same.

~ Subordinates

None.

~ Dependencies

Deletes on TANK_GROUP table.

• Interface

Input parameters:

@group_name (char 30) The name of the tank group to delete.

Returned values:

None.

~ Resources

~ Processing

~ Data

4.9.12 Ip_Delete_SLVL_RDG - Stored Procedure

~ Purpose

Deletes a single record from the SLVL_READING table.

~ Function

Same.

~ Subordinates

None.

~ Dependencies

Deletes on SLVL_READING table.

~ Interface

Input parameters:

@slvl_id (numeric) The SLVL_ID of the row to delete.

Returned values:

None.

~ Resources

~ Processing

~ Data

4.9.13 Ip_Delete_TPTR_RDG - Stored Procedure

~ Purpose

Delete a single row from the TPTR_READING table.

~ Function

Same.

~ Subordinates

None.

~ Dependencies

Delete on TPTR_READING table.

~ Interface

Input parameters:

@tptr_id (numeric) The TPR_ID of the row to delete.

Returned values:

None.

~ Resources

~ Processing

~ Data

4.9.14 Ip_DisSurvAlertsBatchRptILL - Stored Procedure

• Purpose

Displays the ILL readings that exceed baseline limits.

~ Function

Selects ILL readings from the SURV_ALERTS_ILL table that exceed limits. May report on a single tank or all tanks. Uses flags to set filters on the kinds of warnings that will be reported.

~ Subordinates

None.

~ Dependencies

Creates a variety of temporary tables.

Selects from the SURV_ALERTS_ILL table.

- Interface

Input parameters:

@TankName (char 10) The NSTRUCTR_ID of the tank to report on. May have the value "ALL" for all tanks.

@type_0 (char 1) If = 'Y', return the good (between baseline warning limits) values.

@type_1 (char 1) If = 'Y', return the values that have increased > 50% of the way to the upper warning value.

@type_2 (char 1) If = 'Y', return the values that have decreased > 50% of the way to the lower warning value.

@type_3 (char 1) If = 'Y', return the values that have exceeded the upper warning limit.

@type_4 (char 1) If = 'Y', return the values that have fallen below the lower warning limit.

Returned values:

0-many rows of data with the following fields:

convert(char(11), rdg_dttm, 0)

nstructr_id

low_well_no

rdg_value

baseline_value

pass_fail_review

rel_up_lmt

rel_low_lmt

good_values

increase_50

decrease_50

increase_ex

decrease_ex

convert(char(11), run_dttm, 0)

Ordered by nstructr_id, snsr_type_name, rdg_dttm

- Resources
- Processing
- Data

4.9.15 Ip_DisSurvAlertsBatchRptSLVL - Stored Procedure

- Purpose

Displays the surface level readings that exceed baseline limits.

- Function

Selects surface level readings from the SURV_ALERTS_SLVL table that exceed limits. May report on a single tank or all tanks. Uses flags to set filters on the kinds of warnings that will be reported.

- Subordinates

None.

- Dependencies

Makes some temporary tables.

Selects from SURV_ALERTS_SLVL.

- Interface

Input parameters:

@TankName (char 10) The NSTRUCTR_ID of the tank to report on. May have the value "ALL" for all tanks.

@type_0 (char 1) If = 'Y', return the good (between baseline warning limits) values.

@type_1 (char 1) If = 'Y', return the values that have increased > 50% of the way to the upper warning value.

@type_2 (char 1) If = 'Y', return the values that have decreased > 50% of the way to the lower warning value.

@type_3 (char 1) If = 'Y', return the values that have exceeded the upper warning limit.

@type_4 (char 1) If = 'Y', return the values that have fallen below the lower warning limit.

Returned values:

0-many rows of data with the following fields:

convert(char(11), rdg_dttm, 0)

nstructr_id

snsr_type_name

rdg_value

baseline_value

quality_indicator

rel_up_lmt

rel_low_lmt

good_values

increase_50

decrease_50

increase_ex

decrease_ex

convert(char(11), run_dttm, 0)

Ordered by nstructr_id, snsr_type_name, rdg_dttm

- Resources
- Processing
- Data

4.9.16 Ip_DisSurvAlertsBatchRptTPTR - Stored Procedure

- Purpose

Displays the temperature readings that exceed baseline limits.

- Function

Selects temperature readings from the SURV_ALERTS_TPTR table that exceed limits. May report on a single tank or all tanks. Uses flags to set filters on the kinds of warnings that will be reported.

- Subordinates

None.

- Dependencies

Creates a temporary table.

Selects from the SURV_ALERTS_TPTR table.

- Interface

Input parameters:

@TankName (char 10) The NSTRUCTR_ID of the tank to report on. May have the value "ALL" for all tanks.

@type_0 (char 1) If = 'Y', return the good (between baseline warning limits) values.

@type_1 (char 1) If = 'Y', return the values that have increased > 50% of the way to the upper warning value.

@type_2 (char 1) If = 'Y', return the values that have decreased > 50% of the way to the lower warning value.

@type_3 (char 1) If = 'Y', return the values that have exceeded the upper warning limit.

@type_4 (char 1) If = 'Y', return the values that have fallen below the lower warning limit.

Returned values:

0-many rows of data with the following fields:

```
convert(char(11), rdg_dttm, 0)
nstructr_id
comn_name
colctn_idx
rcsys_abbrevd_name
rdg_value
baseline_current
quality_indicator
rel_up_lmt
rel_low_lmt
good_values
increase_50
decrease_50
increase_ex
decrease_ex
convert(char(11), run_dttm, 0)
```

Ordered by nstructr_id, comn_name, colctn_idx, rdg_dttm

- Resources
- Processing
- Data

4.9.17 Ip_Drop_User_SnsrPtsRetParm - Stored Procedure

- Purpose

Delete all entries in the U_SNSR_PTS_RET_PARM table for a given user-ID.

- Function

Same.

- Subordinates

None.

- Dependencies

Deletes on U_SNSR_PTS_RET_PARM.

- Interface

Input parameters:

@Emp_Id (char 7) The employee identification to use when deleting.

Returned values:

None.

- Resources
- Processing
- Data

4.9.18 Ip_GetAllFreqDefnName - Stored Procedure

- Purpose

Return all the frequency names stored within SACS.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from table: `FREQ_NAME`.

- Interface

Input parameters:

None.

Returned values:

A list to frequency names ordered by frequency name.

- Resources
- Processing
- Data

4.9.19 Ip_GetAllSensorPts - Stored Procedure

- Purpose

Accepts the tank name and sensor type and returns the common name of the fixed sample space, the collection sensor index, the abbreviated name of the reading collection system, and the sensor number in the specified sensor type.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

`FIXED_SAMPLE_SPACE`

`ILL_VALUE`

`RDG_COLCTN_SYS_SENSOR`

`READING_COLCTN_SYSTEM`

`SENSOR`

`SENSOR_TYPE`

- Interface

Input parameters:

@TankName (char 30) The name of the tank to report on.

@TypeName (char 30) The type of sensor to report on.

Returned values:

0-many rows with columns:

Common name of fixed sample space.

Collection sensor index.

Abbreviated name of the reading collection system.

Sensor number.

- Resources
- Processing
- Data

4.9.20 Ip_GetAllSensorPtsCommentMaint - Stored Procedure

- Purpose

Given the tank name and sensor type it returns the common name of the fixed sample space, the collection sensor index, and the sensor numbers in the specified sensor type regardless of what readings exist.

- o Function

Same.

- o Subordinates

None.

- o Dependencies

Selects from tables:

FIXED_SAMPLE_SPACE

ILL_VALUE

RDG_COLCTN_SYS_SENSOR

READING_COLCTN_SYSTEM

SENSOR

SENSOR_TYPE

- o Interface

Input parameters:

@TankName (char 30) The name of the tank to report on.

@TypeName (char 30) The type of sensor to report on.

Returned values:

0-many rows with columns:

Common name of fixed sample space.

Collection sensor index.

Sensor number

- o Resources

- o Processing

- o Data

4.9.21 Ip_GetAllSensorTypes - Stored Procedure

- o Purpose

For a given tank returns the different types of sensors that have readings in a tank.

- o Function

Same.

- o Subordinates

None.

- o Dependencies

Selects from:

ILL_VALUE

SENSOR

SLVL_READING

TPTR_READING

- o Interface

Input parameters:

@TankName (char 10) The name of the tank to report on.

Returned values:

0-many SNSR_TYPE_NAMES

- o Resources

- o Processing

- o Data

4.9.22 Ip_GetAllTankGroups - Stored Procedure

- Purpose

Returns all the tank groups known in SACS.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from table: TANK_GROUP:

- ~ Interface

Input parameters:

None.

Returned values:

List of tank group names ordered by tank group name.

- ~ Resources

- ~ Processing

- ~ Data

4.9.23 Ip_GetAllTankNames - Stored Procedure

- ~ Purpose

Returns all the tank and catch-tank names known to SACS.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from table: NAMED_STRUCTURE.

- ~ Interface

Input parameters:

None.

Returned values:

Named structure ordered by named structure.

- Resources

- ~ Processing

- ~ Data

4.9.24 Ip_GetBaselinesILL - Stored Procedure

- Purpose

Returns all the baselines for ILL readings for a given tank, sensor type name, and common fixed sample space name.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from tables:

BASELINES
FIXED_SAMPLE_SPACE
MEASURE
SENSOR_TYPE

- ~ Interface

Input parameters:

@TankName - the name of the tank to report on.
 @SnsrTypeName - the type of sensor to report on.
 @ComnName - the common name of the fixed sample space to report on.

Returned values:

Sensor type name
 Common name of fixed sample space
 Date baseline was entered
 Date baseline started
 Value of the baseline
 Baseline measure name
 Slope rate of baseline
 Relative upper limit
 Relative lower limit

- ~ Resources
- ~ Processing
- ~ Data

4.9.25 Ip_GetBaselinesSLVL - Stored Procedure

- ~ Purpose

Returns all the baselines for surface level readings for a given tank and sensor type name.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from tables:

BASELINES
 FIXED_SAMPLE_SPACE
 MEASURE
 RDG_COLCTN_SYS_SENSOR
 SENSOR
 SENSOR_TYPE

- ~ Interface

Input parameters:

@TankName - the name of the tank to report on.
 @SnsrTypeName - the type of sensor to report on.

Returned values:

Sensor type name
 Date baseline was entered
 Date baseline started
 Value of the baseline
 Baseline measure name
 Slope rate of baseline
 Relative upper limit
 Relative lower limit

- ~ Resources
- ~ Processing
- ~ Data

4.9.26 Ip_GetBaselinesTPTR - Stored Procedure

- Purpose

Returns all the baselines for temperature readings for a given tank, sensor type name, and index number.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

BASELINES
FIXED_SAMPLE_SPACE
MEASURE
RDG_COLCTN_SYS_SENSOR
SENSOR
SENSOR_TYPE

- Interface

Input parameters:

@TankName - the name of the tank to report on.
@ComnName - the anem of the common fixed sample space to report on.
@Colctnidx - the index number of the temperature sensor to report on.

Returned values:

Common name of fixed sample space
Collection index
Date baseline was entered
Date baseline started
Date baseline ended
Value of the baseline
Baseline measure name
Baseline dtm
Slope rate of baseline
Slope rate measure name
Relative upper limit
Relative lower limit

- Resources
- Processing
- Data

4.9.27 Ip_GetBatchTime - Stored Procedure

- Purpose

Returns the distinct run times for the surveillance alert report rows.

- Function

Selects the distinct RUN_DTM from the SURV_ALERTS_TPTR, SURV_ALERTS_SLVL, and SURV_ALERTS_ILL tables and performs a UNION.

- Subordinates

None.

- Dependencies

Select from tables:

SURV_ALERTS_ILL
SURV_ALERTS_SLVL
SURV_ALERTS_TPTR

- Interface

Input parameters:

None.

Returned values:

The date-time that the surveillance alert reports were run.

- Resources
- Processing
- Data

4.9.28 Ip_GetCurrentGoodSLVL - Stored Procedure

- Purpose

To get the latest surface level value for a tank that has been marked good.

- Function

Selects the latest surface level for a tank where the surface level reading has been marked Good. Then returns that date-time and the value of the reading.

- Subordinates

None.

- Dependencies

Selects from SLVL_READING.

- Interface

Input parameters:

@tank (char 10) The name of the tank to report on.

Returned values:

The latest SLVL_QTY and the SLVL_DTM for it.

- Resources
- Processing
- Data

4.9.29 Ip_GetDistMeasName - Stored Procedure

- Purpose

Get all the unit of measure that relate to distance.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from table MEASURE.

- Interface

Input parameters:

None.

Returned values:

The MEAS_NAME for measures that have to do with distance.

- Resources
- Processing
- Data

4.9.30 Ip_GetFreqVtoSnsrGrps - Stored Procedure

- Purpose

Returns the distinct sensor reading groups from the FSS_FREQUENCY table.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from table: FSS_FREQUENCY.

- Interface

Input parameters:

None.

Returned values:

Distinct values of SNSR_RDG_GROUP in the FSS_FREQUENCY table.

- Resources
- Processing
- Data

4.9.31 Ip_GetFSSSTank - Stored Procedure

- Purpose

For a given tank, returns the distinct fixed sample space common names, sensor reading groups, and, if available, the frequency names for that tank.

- Function

Same.

- Subordinates

None.

- Dependencies

Select from table:

FIXED_SAMPLE_SPACERDG_COLCTN_SYS_SENSOR
FREQUENCY_DEFN
FSS_SAMPLE_FREQ
RDG_COLCTN_SYS_SNSR_TYPE
SENSOR
SENSOR_TYPE

- Interface

Input parameters:

@tank (char 10) The name of the tank to report on.

Returned values:

0-many:

Common name of fixed sample space
Sensor reading group
Frequency Name

- Resources
- Processing
- Data

4.9.32 Ip_GetFreqDefn - Stored Procedure

- Purpose

Given a frequency name, it returns the corresponding row from the FREQUENCY_DEFN table.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

FREQUENCY_DEFN

MEASURE

Interface

Input parameters:

@freq_name (char 30) The name of the frequency definition to return.

Returned values:

Limit

Measure Code for Limit

Warning Value

Measure Code for Warning Value

Extension Value

Measure Code for Extension Value

Resources

Processing

Data

4.9.33 Ip_GetFreqVioRptFlt - Stored Procedure

@e_flag (char 1)

@l_flag (char 1)

@w_flag (char 1)

@snsr_type (char 15)

4.9.34 Ip_GetManualSensorTypes_SLVL - Stored Procedure

Purpose

For a given tank, returns the sensor type name and the sensor number for sensors that have their data collected manually. E.g., the reading collection system number = 2.

Function

Same.

Subordinates

None.

Dependencies

Select from tables:

RDG_COLCTN_SYS_SENSOR

SENSOR

Interface

Input parameters:

@TankName (char 10) The name of the tank to examine.

Returned values:

0-many:

Sensor type name

Sensor Number

Resources

Processing

Data

4.9.35 Ip_GetMessages - Stored Procedure

Purpose

Get the messages from the VB_MESSAGE_OTD table.

Function

Get the messages from the VB_MESSAGE_OTD table that are current; e.g. the date range in the message covers today.

Subordinates

None.

- Dependencies

Selects from table: VB_MESSAGE_OTD

- Interface

Input parameters:

None.

Returned values:

GROUP_NAME - the name of the group that sees the message.

MESSAGE - the message of the day.

- Resources
- Processing
- Data

4.9.36 Ip_GetINSTRUCTRComments - Stored Procedure

- Purpose

Adds rows to the U_COMMENT_REPORT table that pertain to Named Structure comments for a given tank over a date range.

- Function

Same.

- Subordinates

None.

- Dependencies

Insert into: U_COMMENT_REPORT.

Select from NS_COMMENT.

- Interface

Input parameters:

@nstructr_id (char 30) The name of the tank to report on.

@start_dttm (datetime) The start of the time period to examine.

@end_dttm (datetime) The end of the time period to examine.

Returned values:

None returned directly.

The U_COMMENT_REPORT receives:

Comment date-time

Named structure ID

Named Structure comment text

'Tank Comments' string

Ordered by date-time.

- Resources
- Processing
- Data

4.9.37 Ip_GetNamedStructureComment - Stored Procedure

- Purpose

Returns the comments for a named structure that lie within a date range.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from NS_COMMENT.

- Interface

Input parameters:

- @NamedStructure (char 10) The name of the tank to report on.
- @BeginDate (datetime) The start date-time for comments.
- @EndDate (datetime) The end date-time for comments.

Returned values:

0-many rows of:

- Named Structure Comment ID
- Named Structure Comment Date-time
- Named Structure Comment Text

Ordered by Named Structure Comment Date-time

- Resources
- Processing
- Data

4.9.38 Ip_GetRDGComments - Stored Procedure

- Purpose

Fills in the U_COMMENT_REPORT with reading comments for a given tank, sensor type, and date range. Only one type of reading comment is stored per invocation.

- Function

Same.

- Subordinates

None.

- Dependencies

Select from tables:

- SENSOR
- SENSOR_TYPE
- SLVL_READING
- TPTR_READING.

Inserts into:

U_COMMENT_REPORT.

- Interface

Input parameters:

- @nstructr_id (char 30) Tank to report on.
- @sensor_no (Int) Sensor to report in (used in SLVL and TPTR comments).
- @sensor_type (char 30)
- @start_dttm (datetime) Start of date-time for comments.
- @end_dttm (datetime) End of date-time for comments.

Returned values:

None.

Inserts into U_COMMENTS_REPORT:

- Comment date-time
- Sensor type name
- Reading Comment Text
- 'Reading Comments', a string

- Resources
- Processing
- Data

4.9.39 Ip_GetSNSRComments - Stored Procedure

- Purpose

Fills in the U_COMMENTS_REPORT table with sensor comments for a given tank, sensor type, and date range.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

SENSOR_TYPE

Insert into:

U_COMMENTS_REPORT.

- Interface

Input parameters:

@nstructr_id (char 30) Tank to report on.

@sensor_no (int) Sensor to report in (used in SLVL and TPTR comments).

@sensor_type (char 30)

@start_dttm (datetime) Start of date-time for comments.

@end_dttm (datetime) End of date-time for comments.

Returned values:

None.

Inserts into U_COMMENTS_REPORT:

Comment date-time

Sensor type name

Reading Comment Text

'Reading Comments', a string

- Resources

- Processing

- Data

4.9.40 Ip_GetSensorComment - Stored Procedure

- Purpose

Retrieve the comments for a sensor where the comments lie in a given date range.

- Function

Same.

- Subordinates

None.

- Dependencies

Select from S_COMMENT table.

- Interface

Input parameters:

@Sensor (int) The sensor number to report on.

@BeginDate (datetime) Beginning date-time to report on.

@EndDate (datetime) Ending date-time to report on.

Returned values:

0-many:

Comment ID

Comment Date-time

Comment text

- ~ Resources
- Processing
- ~ Data

4.9.41 Ip_GetSensorGroupBySensorType - Stored Procedure

- ~ Purpose

Returns the sensor reading group when provided with the sensor type name.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from SENSOR_TYPE table.

- ~ Interface

Input parameters:

@snsr_type (char 30) The sensor type to use.

Returned values:

Sensor type name

- ~ Resources
- ~ Processing
- ~ Data

4.9.42 Ip_GetSensorNumber - Stored Procedure

- Purpose

Given a tank and a sensor type it returns the sensor number. Only good for surface level sensors where there is a single sensor for a given type in a tank. Probably won't work for thermocouple trees...

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from tables:

RDG_COLCTN_SYS_SENSOR.

SENSOR

- ~ Interface

Input parameters:

@Tank (char 10) The tank to report on.

@SnsrType (char 30) The type of sensor to report on.

Returned values:

Sensor number.

- ~ Resources
- ~ Processing
- ~ Data

4.9.43 Ip_GetSensorPts - Stored Procedure

- ~ Purpose

Get a list of sensor points when given the named structure (tank), date range, and sensor type name.

o Function

Same.

o Subordinates

None.

o Dependencies

Select from tables:

FIXED_SAMPLE_SPACE

ILL_VALUE

SENSOR

SENSOR_TYPE

SLVL_READING

TPTR_READING

o Interface

Input parameters:

@TankName (char 30)

@BegDate (datetime)

@EndDate (datetime)

@TypeName (char 30)

Returned values:

Varies by sensor type

T Fixed sample space common name

T Collection sensor index

T Reading Collection System number

T Sensor number

T Sensor reading group name

S Fixed sample space common name

S Collection sensor index

S Abbreviated name (of what??)

S Reading Collection System number

S Sensor number

S Sensor reading group name

I Sensor type name

I NULL

I LOW well number

I 1 (a place holder)

I Sensor reading group name

o Resources

o Processing

o Data

4.9.44 Ip_GetSensorTypes - Stored Procedure

o Purpose

Returns a list of sensor type that have at least 1 reading for a specified tank with a specified date range.

o Function

Same.

o Subordinates

None.

o Dependencies

Select on tables

ILL_VALUE

SLVL_READING

TPTR_READING

- Interface

Input parameters:

@TankName (char 10) The named structure (tank) to report on.

@BegDate (datetime) The starting date to scan.

@EndDate (datetime) The ending date to scan.

Returned values:

0-many:

Sensor number (temperature and surface level sensors)

Sensor type name (ILL)

- Resources
- Processing
- Data

4.9.45 Ip_GetSensorTypes_GST - Stored Procedure

- Purpose

Given a tank and a date range, returns the sensor types and whether the suspect data status code = G, S, or T. Any or all the suspect data status codes may be checked at the same time.

- Function

Same.

- Subordinates

None.

- Dependencies

Select on tables:

ILL_VALUE

SLVL_READING

TPTR_READING

- Interface

Input parameters:

@TankName (char 10) Tank to report on.

@BegDate (datetime) Beginning date to search on.

@EndDate (datetime) Ending date to search on.

@ChkGood (bit 1) If set, check for good readings.

@ChkSuspect (bit 1) If set, check for suspect readings.

@ChkTranscribed (bit 1) If set, check for transcribed readings.

Returned values:

0-many Sensor Type Names for the temperature, surface level, and ILL sensors that have readings in the date range.

- Resources
- Processing
- Data

4.9.46 Ip_GetSensorTypes_SLVL - Stored Procedure

- Purpose

Returns a list of surface level sensor types and sensor numbers as configured in the database. It does NOT check whether readings exist for these sensors.

- Function

Same.

- Subordinates

None.

- Dependencies

Select from tables:

RDG_COLCTN_SYS_SENSOR
SENSOR

- Interface

Input parameters:

@TankName (char 10) Tank name to search for.

Returned values:

0-many:

Sensor type name
Sensor number

- Resources
- Processing
- Data

4.9.47 Ip_GetSnrPtsILL - Stored Procedure

- Purpose

Returns a list of sensor type names for ILLs for a given tank.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

FIXED_SAMPLE_SPACE
RDG_COLCTN_SYS_SENSOR
SENSOR_TYPE

- Interface

Input parameters:

@TankName (char 10) Tank to return values for.

Returned values:

0-many

Sensor type name
Fixed Sample Space common name

- Resources
- Processing
- Data

4.9.48 Ip_GetSnrPtsSLVL - Stored Procedure

- Purpose

Returns a list of surface level sensors in a tank.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

RDG_COLCTN_SYS_SENSOR

SENSOR
SENSOR_TYPE

- Interface
- Input parameters:
 - @TankName (char 10) Tank to return values for.
- Returned values:
 - 0-many Sensor Type names.
- Resources
- Processing
- Data

4.9.49 Ip_GetSnsrPtsTPTR - Stored Procedure

- Purpose
- Returns the type of temperature sensors configured for a tank.
- Function
- Same.
- Subordinates
- None.
- Dependencies
- Selects from tables:
 - FIXED_SAMPLE_SPACE
 - RDG_COLCTN_SYS_SENSOR
 - SENSOR
- Interface
- Input parameters:
 - @TankName (char 10) (char 10) Tank to return values for.
- Returned values:
 - 0-many Sensor Type names.
- Resources
- Processing
- Data

4.9.50 Ip_GetSnsrRdgGroup - Stored Procedure

- Purpose
- Returns the sensor reading groups for which there are readings in a tank, or all tanks, for a date range.
- Function
- Same.
- Subordinates
- None.
- Dependencies
- Selects from tables:
 - ILL_VALUE
 - RDG_COLCTN_SYS_SENSOR
 - SENSOR
 - SENSOR_TYPE
 - SLVL_READING
 - TPTR_READING
- Interface
- Input parameters:

@TankName (char 10) Tank name to search, or the value 'ALL' for all tanks.

@BegDate (datetime) Beginning date to look for.

@EndDate (datetime) End date to look for.

Returned values:

0-many Sensor reading groups.

- Resources
- Processing
- Data

4.9.51 Ip_GetSnsrRdgGroupByTank - Stored Procedure

• Purpose

Return the sensor reading groups configured for 1 tank. Does not check to see if there are readings present for this group.

• Function

Same.

• Subordinates

None.

• Dependencies

Selects from tables:

RDG_COLCTN_SYS_SNSR_TYPE

SENSOR_TYPE

• Interface

Input parameters:

@TankName (char 10) The name of the tank to report on.

Returned values:

0-many Sensor reading groups present in the tank.

- Resources
- Processing
- Data

4.9.52 Ip_GetSnsrSpacing - Stored Procedure

• Purpose

Returns the sensor spacing in inches for a given collection number.

• Function

Same.

• Subordinates

None.

• Dependencies

Selects from tables:

COLCTN_SNSR_SPACING_SPEC

COLLECTION

• Interface

Input parameters:

@colctn_no (smallint) The collection to report on.

Returned values:

0-many

Sensor index number

Vertical offset from bottom of tank (Z quantity + Z offset quantity)

- Resources
- Processing

- ~ Data

4.9.53 Ip_GetSurvAlertsBatchRpt - Stored Procedure

- ~ Purpose

Calls a number of other stored procedures to fill the Surveillance Limit Report tables.

- ~ Function

Same.

- ~ Subordinates

Calls: Ip_RetrieveTPTRDataBatch
 Ip_RetrieveSLVLDDataBatch
 Ip_RetrieveILLDataBatch

- ~ Dependencies

None.

- Interface

Input parameters:

@DaysBack (smallint) The number of days before present to scan to out-of-tolerance values.

Returned values:

None.

- ~ Resources

- ~ Processing

- ~ Data

4.9.54 Ip_GetSurvAlertsRptILL - Stored Procedure

- ~ Purpose

Create the Surveillance Report rows for the surveillance alert report for 1 or all tanks for the ILL readings. Stores results in a #temporary table. Creates reports on demand - this isn't the batch job.

- ~ Function

Same. The output for the ILL value is in inches - there are some internal conversions to ensure this.

- ~ Subordinates

None.

- ~ Dependencies

Selects on tables:

BASELINES
 ILL_VALUE
 MEAS_CONV_N
 SENSOR_TYPE

- ~ Interface

Input parameters:

@TankName (char 10) The tank name to analyze. If the string 'ALL' it will analyze all tanks.

@BegDate (datetime) Beginning date to search for out-of-tolerance values.

@EndDate (datetime) Ending date to search for out-of-tolerance values.

@SnsrRdgGroup (char 15) Passed in but not used at all in code! Looks like C*m*ll was here.

@type_0 (char 1) If 'Y', report on values that are near the baseline.

@type_1 (char 1) If 'Y', report on values > 50% of the way to the upper baseline.

@type_2 (char 1) If 'Y', report on values > 50% of the way to the lower baseline.

@type_3 (char 1) If 'Y', report on values above the upper baseline.

@type_4 (char 1) If 'Y', report on values below the lower baseline.

Returned values:

0-many rows of:

Reading date-time (as char(11))

Named Structure ID

LOW Well Number

Sensor type name

Reading value (as char(7,2))

Current baseline value (as char(7,2))

Pass-fail review

Relative upper limit (as char(7,2))

Relative lower limit (as char(7,2))

Flags:

Good Values (as char(4,0))

Increase > 50% (as char(4,0))

Decrease > 50% (as char(4,0))

Increase Exceeded (as char(4,0))

Decrease Exceeded (as char(4,0))

Date-time the report was run

- Resources
- Processing
- Data

4.9.55 Ip_GetSurvAlertsRptSLVL - Stored Procedure

- Purpose

Create the Surveillance Report rows for the surveillance alert report for 1 or all tanks for the surface level readings. Stores results in a #temporary table. Creates reports on demand - this isn't the batch job.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

BASELINES

FIXED_SAMPLE_SPACE

RDG_COLCTN_SYS_SENSOR

READING_COLCTN_SYSTEM

SENSOR

SENSOR_TYPE

SLVL_READING

- Interface

Input parameters:

@TankName (char 10) The tank name to analyze. If the string 'ALL' it will analyze all tanks.

@BegDate (datetime) Beginning date to search for out-of-tolerance values.

@EndDate (datetime) Ending date to search for out-of-tolerance values.

@SnsrRdgGroup (char 15) Passed in but not used at all in code! Looks like C*m"ll was here.

@type_0 (char 1) If 'Y', report on values that are near the baseline.

- @type_1 (char 1) If 'Y', report on values > 50% of the way to the upper baseline.
- @type_2 (char 1) If 'Y', report on values > 50% of the way to the lower baseline.
- @type_3 (char 1) If 'Y', report on values above the upper baseline.
- @type_4 (char 1) If 'Y', report on values below the lower baseline.

Returned values:

0-many rows of:

Reading date-time (as char(11))
 Named Structure ID
 Sensor type name
 Reading value (as char(7,2))
 Current baseline value (as char(7,2))
 Suspect data status code
 Relative upper limit (as char(7,2))
 Relative lower limit (as char(7,2))
 Flags:
 Good Values (as char(4,0))
 Increase > 50% (as char(4,0))
 Decrease > 50% (as char(4,0))
 Increase Exceeded (as char(4,0))
 Decrease Exceeded (as char(4,0))

Date-time the report was run

- Resources
- Processing
- Data

4.9.56 Ip_GetSurvAlertsRpTPTR - Stored Procedure

- Purpose

Create the Surveillance Report rows for the surveillance alert report for 1 or all tanks for the temperature readings. Stores results in a #temporary table. Creates reports on demand - this isn't the batch job.

- Function

Same.

- Subordinates

None.

- Dependencies

BASELINES
 FIXED_SAMPLE_SPACE
 RDG_COLCTN_SYS_SENSOR
 READING_COLCTN_SYSTEM
 SENSOR
 SENSOR_TYPE
 TPTR_READING

- Interface

Input parameters:

- @TankName (char 10) The tank name to analyze. If the string 'ALL' it will analyze all tanks.
- @BegDate (datetime) Beginning date to search for out-of-tolerance values.
- @EndDate (datetime) Ending date to search for out-of-tolerance values.
- @SnsRdgGroup (char 15) Passed in but not used at all in code! Looks like C*m*ll was here.
- @type_0 (char 1) If 'Y', report on values that are near the baseline.

- @type_1 (char 1) If 'Y', report on values > 50% of the way to the upper baseline.
- @type_2 (char 1) If 'Y', report on values > 50% of the way to the lower baseline.
- @type_3 (char 1) If 'Y', report on values above the upper baseline.
- @type_4 (char 1) If 'Y', report on values below the lower baseline.

Returned values:

0-many rows of:

Reading date-time (as char(11))
 Named Structure ID
 Fixed Sample Space common name
 Collection Index number
 Reading collection system abbreviated name
 Reading value (as char(7,2))
 Current baseline value (as char(7,2))
 Suspect data status code
 Relative upper limit (as char(7,2))
 Relative lower limit (as char(7,2))

Flags:

Good Values (as char(4,0))
 Increase > 50% (as char(4,0))
 Decrease > 50% (as char(4,0))
 Increase Exceeded (as char(4,0))
 Decrease Exceeded (as char(4,0))

Date-time the report was run

- Resources
- Processing
- Data

4.9.57 Ip_GetTankAndSensorComment - Stored Procedure

- Purpose

Retrieve the tank and sensor comments for the V&V Grid double click event.

- Function

Return a list of the sensor and tank comments for the tank where the person is working. Sensor comments are returned before tank comments. The comments are sorted chronologically.

- Subordinates

None.

- Dependencies

Selects from tables:

NS_COMMENT
 RDG_COLCTN_SYS_SENSOR
 S_COMMENT
 U_SNSR_PTS_RET_PARM

- Interface

Input parameters:

@Emp_Id (EMP_ID 16) The ID to use when searching in the U_SNSR_PTS_RET_PARM table for the tank that this person is working with.

Returned values:

0-many rows of:

Date-time of comment (as char(12))
 Comment text

- Resources

- Processing
- Data

4.9.58 Ip_GetTankAndSensorCommentCnt - Stored Procedure

- Purpose

This will return the count of tank and sensor comments for a given tank and date-time range. Used with the V&V Grid form.

- Function

Counts the number of relevant comments in the S_COMMENT and NS_COMMENT tables. Uses the U_SNSR_PTS_RET_PARM table to find which tank and date range to report.

- Subordinates

None.

- Dependencies

Selects from tables:

NS_COMMENT
RDG_COLCTN_SYS_SENSOR
S_COMMENT
U_SNSR_PTS_RET_PARM

- Interface

Input parameters:

@Emp_id (EMP_ID 16) The ID of the user in the U_SNSR_PTS_RET_PARM table.

Returned values:

A count of the comments available.

- Resources
- Processing
- Data

4.9.59 Ip_GetTankGroupDesc - Stored Procedure

- Purpose

Returns the description for a given tank group.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from TANK_GROUP table.

- Interface

Input parameters:

@group_name (char 30) The name fo the tank group to report on.

Returned values:

The description of the tank group.

- Resources
- Processing
- Data

4.9.60 Ip_GetTanksInGroup - Stored Procedure

- Purpose

Returns a list of all tanks in a given tank group.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from TANK_IN GROUP table.

- Interface

Input parameters:

- @group_name (char 30) The name of the tank group to report on.

Returned values:

0-many Named Structure IDs (tanks)

- Resources

- Processing

- Data

4.9.61 Ip_GetTankName - Stored Procedure

- Purpose

Returns the tanks known to SACS.

- Function

Selects all the named structures from NAMED_STRUCTURE that have a named structure type name of 'Tank'.

- Subordinates

None.

- Dependencies

Selects from table: NAMED_STRUCTURE.

- Interface

Input parameters:

None

Returned values:

List of tanks in the SACS database.

- Resources

- Processing

- Data

4.9.62 Ip_GetTimeMeasCode - Stored Procedure

- Purpose

Return the codes for measure that have to do with Time.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from table: MEASURE.

- Interface

Input parameters:

None.

Returned values:

The measure codes that have to do with time.

- Resources

- Processing

- Data

4.9.63 Ip_GetTpTrMeasName - Stored Procedure

- Purpose

Returns the names for the units of measure that have to do with Temperature.

- Function

Same.

- Subordinates

None.

- Dependencies

Select from table: MEASURE.

- Interface

Input parameters:

None.

Returned values:

The measure name for measure that have to do with Temperature.

- Resources

- Processing

- Data

4.9.64 Ip_GetVersion - Stored Procedure

- Purpose

Gets the Visual Basic version code.

- Function

Selects the VB_VERSION_TEXT from VB_VERSION.

- Subordinates

None.

- Dependencies

Select from table: VB_VERSION.

- Interface

Input parameters:

None.

Returned values:

Text of the string stored in VB_VERSION table.

- Resources

- Processing

- Data

4.9.65 Ip_Get_COLCTN_NO - Stored Procedure

- Purpose

Returns the collection numbers for a given tank and date range for which there are readings.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

RDG_COLCTN_SYS_SENSOR

TPTR_READING

- Interface

Input parameters:

@tank (char 10) Tank to report on.

@start_dttm (datetime) Start of time to look for readings.

@end_dttm (datetime) End of time to look for readings.

Returned values:

0-many Collection Numbers

- Resources
- Processing
- Data

4.9.66 Ip_get_farm - Stored Procedure

- Purpose

Returns the farms known to SACS.

- Function

Selects all of the farms from the NSTRUCTR_ID table (those with a named structure type = 2).

- Subordinates

None.

- Dependencies

Selects from table NSTRUCTR_ID.

- Interface

Input parameters:

None.

Returned values:

List of the named structure IDs.

- Resources
- Processing
- Data

4.9.67 Ip_Get_MRDCNAM_NAME - Stored Procedure

- Purpose

Returns a sensor type name and manual reading device common name for a given tank and fixed sample space for a temperature reading at a certain time.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

MAN_RDG_DEV_COMN_NAME
RDG_COLCTN_SYS_SENSOR
RDG_DEVICE
RDGDEV_VALID_SENSOR
TPTR_READING

- Interface

Input parameters:

@nstructr_id (char 10) Tank to report on.

@fss_no (smallint) The fixed sample space to report on.

@dttm (datetime) The date-time of a reading to report on.

Returned values:

0-many:

Sensor type names

Manual Reading Device names

- Resources
- Processing
- Data

4.9.68 Ip_Get_MRDCNAM_NAME_byTank - Stored Procedure

- Purpose

Given a tank, fixed sample space type, and date-time this proc returns the sensor type names, manual reading device names, and reading device numbers.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

```
MAN_RDG_DEV_COMN_NAME
RDG_COLCTN_SYS_SENSOR
RDG_DEVICE
RDGDEV_VALID_SENSOR
TPTR_READING
```

- Interface

Input parameters:

@nstructr_id (char 10) The tank to report on.

@fss_type_no (smallint 2) The type of fixed sample space to report on.

@fss_no (smallint 2) The fixed sample space to report on.

@dtm (datetime) The date-time of the temperature reading to report on.

Returned values:

0-many

Sensor type name

Manual Reading Device name

- Resources
- Processing
- Data

4.9.69 Ip_Get_Num_TPTR_Risers - Stored Procedure

- Purpose

Return the number of risers configured for all tanks in a given tank farm.

- Function

Returns the number of Risers (type 7), Liquid Observation Wells (type 8), and Embedded in Structure (type 4) for all tanks in a tank farm.

- Subordinates

None.

- Dependencies

Selects from tables:

```
FIXED_SAMPLE_SPACE
TANK_IN_FARM_V
```

- Interface

Input parameters:

@farm (char 10) The farm to report on.

Returned values:

The number of risers in the tank farm.

- Resources
- Processing
- Data

4.9.70 Ip_Get_Num_TPTR_Risers_byTank - Stored Procedure

- Purpose

Given a tank name returns a list of the risers for that tank.

- Function

Returns the number of Risers (type 7), Liquid Observation Wells (type 8), Embedded in Structure (type 4), and historical (type 9) for a given tank.

- Subordinates

None.

- Dependencies

Selects from tables:

FIXED_SAMPLE_SPACE

TPTR_READING

RDG_COLCTN_SYS_SENSOR

- Interface

Input parameters:

@tank (char 10) Tank to report on.

@dtm (datetime) Date-time of reading to report on.

Returned values:

Count of the risers in a tank.

- Resources
- Processing
- Data

4.9.71 Ip_Get_Num_TPTR_Sensors - Stored Procedure

- Purpose

Returns the number of temperature sensors that have readings in a given tank for a date-time range.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from tables:

RDG_COLCTN_SYS_SENSOR

TPTR_READING

- Interface

Input parameters:

@structr_id (char 10) Tank to report on.

@start_dtm (datetime) Start of time to look at.

@end_dtm (datetime) End of time to look at.

Returned values:

Maximum of COLCTN_SNSR_IDX for the tank.

- Resources
- Processing
- Data

4.9.72 Ip_Get_RDG_DEVICE_NAME - Stored Procedure

- Purpose
Returns a list of reading device common names for a given sensor type name.
- Function
Same.
- Subordinates
None.
- Dependencies
Selects from Tables:
RDG_DEVICE
RDGDEV_VALID_SENSOR
- Interface
Input Parameters:
@snsr_type (char 30) Type of sensor to report on.
- Returned values:
Distinct values for RDGDEV_NAME, RDGDEV_NO.
- Resources
- Processing
- Data

4.9.73 Ip_Get_TPTR_Data - Stored Procedure

- Purpose
Returns the temperature quantity, ID, and comment for the chosen sensor number and date-time.
- Function
Same.
- Subordinates
None.
- Dependencies
Selects from Tables:
READING_COLCTN_SYSTEM
TPTR_READING
- Interface
Input Parameters:
@snsr_no (Int) Sensor number to report on.
@dtm (datetime) Date time to report on.
- Returned values:
Temperature quantity, reading collection system number, abbreviated name, measurement name, temperature reading comment, temperature ID.
- Resources
- Processing
- Data

4.9.74 Ip_Get_TPTR_RDGS - Stored Procedure

- Purpose
Return all the temperature readings for a given tank and date range.
- Function
Same.
- Subordinates
None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
MEAS_CONV_N_V
RDG_COLCTN_SYS_SENSOR
READING_COLLECTION_SYSTEM
SENSOR
TPTR_READING

- Interface

Input Parameters:

@nstructr_id (char 10) Tank to report on.
@end_dttm (datetime) Last date to report on.
@start_dttm (datetime) First date to start on.

Returned values:

Temperature date-time, named structure ID, common name, temperature quantity, collection sensor index, sensor type name, fixed sample space number, fixed sample space type number, collection number, reading collection system number, suspect data status code, abbreviated name, measure number.

- Resources
- Processing
- Data

4.9.75 Ip_Get_TPTR_Risers - Stored Procedure

- Purpose

Returns a list of tanks in a given farm which have a fixed sample space type of 'Riser' or 'Embedded In Structure'.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
TANK_IN_FARM_V

- Interface

Input Parameters:

@farm (char 10) Farm to report on.

Returned values:

Named structure, common name of fixed sample space, fixed sample space number and type number.

- Resources
- Processing
- Data

4.9.76 Ip_Get_TPTR_Risers_byTank - Stored Procedure

- Purpose

Returns the list of risers for the given tank for a date-time for which there is a temperature reading.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
RDG_COLCTN_SYS_SENSOR
TPTR_READING

- Interface

Input Parameters:

@tank (char 10) Tank to report on.
@dtm (datetime) Date

Returned values:

Tank name, common name, fixed sample space type number and number.

- Resources

- Processing

- Data

4.9.77 Ip_Get_TPTR_Sensors - Stored Procedure

- Purpose

Returns a list of sensor number for a given tank/fixed sample space. It assumes that the collection system is manual.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

RDG_COLCTN_SYS_SENSOR

- Interface

Input Parameters:

@fss_no (smallint) Fixed sample space ID to report on.
@nstructr_id (char 10) Tank to report on.
@fss_type_no (smallint) Type of fixed sample space to report on.

Returned values:

Sensor index number, sensor number, collection number.

- Resources

- Processing

- Data

4.9.78 Ip_Get_TPTR_Sensors_byTank - Stored Procedure

- Purpose

Returns the sensor number, index, and collection number for a given tank and fixed sample space..

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

RDG_COLCTN_SYS_SENSOR

- Interface

Input Parameters:

@nstructr_id (char 10) Tank to report on.

@fss_no (smallint) Fixed sample space number to report on.

@fss_type_no (tinyint) Type of fixed sample space to report on.

Returned values:

Collection sensor index, sensor number, collection number.

- Resources
- Processing
- Data

4.9.79 Ip_Get_TPTR_Time - Stored Procedure

- Purpose

Returns the times for temperature reading taken in the given tank on the given day.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

TPTR_READING

- Interface

Input Parameters:

@nstructr_id (char 10) Tank to report on.

@dtm (datetime) Date to report on.

Returned values:

Times for values converted to hh:miss format.

- Resources
- Processing
- Data

4.9.80 Ip_Get_User_SnsrPtsRetParm - Stored Procedure

- Purpose

Retrieves values from the U_SNSR_PTS_RET_PARM table for a certain user. Generally used as the last step in a delete - load - report cycle which employs U_SNSR_PTS_RET_PARM as the repository for parameters for a retrieval.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE

ILL_VALUE

RDG_COLCTN_SYS_SENSOR

SENSOR_TYPE

SLVL_READING

TPTR_READING

U_SNSR_PTS_RET_PARM

- Interface

Input Parameters:

@Emp_Jd (char 7) ID of the employee working with the table.

Returned values:

The values returned depend on the type of data selected

TEMPERATURE

Date-time, reading collection system number, sensor number, temperature quantity, suspect data status code, employee ID, common name, collection sensor index, measure number, abbreviated date-time, and temperature comment.

SURFACE LEVEL

Date-time, reading collection system number, sensor number, surface level quantity, suspect data status code, employee ID, sensor type name, NULL, abbreviated date-time, and surface level comment.

ILL

Date-time, login sequence number, LOW well number, corrected ILL value, pass-fail review, revised employee ID, NULL, NULL, abbreviated date-time, and ILL comment.

- ~ Resources
- ~ Processing
- ~ Data

4.9.81 Ip_InsertBaseline - Stored Procedure

- Purpose

Receives all the things that are needed for entering a baseline record.

- ~ Function

Same.

- ~ Subordinates

None.

- ~ Dependencies

Selects from Tables:

BASELINES

MEASURE

- ~ Interface

Input Parameters:

@TankName (char 10) The name of the tank we are working in.

@SnsrTypeName (char 30) Type of sensor to which the baseline applies.

@SnsrNo (Int) Sensor number for this baseline.

@StartDttm (datetime) Starting date and time for the baseline.

@BaselineValue (real 4) Value for the baseline.

@BaselineDttm (datetime) Date-time that the baseline starts, e.g., when the slope offset is zero.

@BslnMeasName (char 40) Unit of measure for the baseline.

@SlopeRate (real 4) Value of the slope.

@SlopeMeasName (char 40) The unit of measure for the slope. Usually a rate.

@RelUpLmt (real 4) Upper limit relative to the baseline.

@RelLowLmt (real 4) Lower limit relative to the baseline.

@EmpId (char 7) Employee who entered the data

Returned values:

None.

- ~ Resources
- ~ Processing

- Data

4.9.82 Ip_InsertFSSSRow - Stored Procedure

- Purpose

Receives data from the client to create/modify a row in the FSS_FREQUENCY table. The stored procedure will update an existing row or add a new one.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
FREQUENCY_DEFN
FSS_FREQUENCY
NAMED_STRUCTURE

- Interface

Input Parameters:

@tank_name (char 10) The tank where the sampling frequency is to be assigned.

@fss_comn_name (char 20) The common name of the fixed sampling space.

@snsr_rdg_group (char 15) The type of sensor that will be assigned a frequency.

@freq_name (char 30) The name of the frequency to assign.

Returned values:

None.

- Resources
- Processing
- Data

4.9.83 Ip_InsertFreqDefn - Stored Procedure

- Purpose

Used by the client to modify or insert a row in the Frequency Definition table.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

FREQUENCY_DEFN
MEASURE

- Interface

Input Parameters:

@freq_name (char 15) The frequency name to insert/modify.

@limit (real) The value for the frequency limit.

@limit_meas_code (char 10) The unit of measure for the limit.

@warning_value (real) The value for the warning limit.

@warn_meas_code (char 10) The unit of measure for the warning limit.

@extension_value (real 4) The value for the extended warning limit.

@extension_meas_code (char 10) The unit of measure for the extended warning limit.

Returned values:

None.

- Resources
- Processing
- Data

4.9.84 Ip_InsertNamedStructureComment - Stored Procedure

- Purpose

Add a comment to the Named Structure Comment table.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

NS_COMMENT

- Interface

Input Parameters:

@NamedStructure (char 10) The named structure to comment on.

@Comment_Text (varchar 250) The text of the comment.

@Comment_DTM (datetime) The date-time for the comment.

Returned values:

None.

- Resources
- Processing
- Data

4.9.85 Ip_InsertSLVL - Stored Procedure

- Purpose

Adds a new row to the surface level reading table. It relies on the triggers and defaults in the SLVL_READING table to add certain values to the row.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

SLVL_READING

- Interface

Input Parameters:

@rcsys_no (smallint) Reading collection system for reading.

@snsr_no (int) Sensor number for reading.

@dtm (datetime) Date and time of reading.

@nstructr_id (char 10) Tank where reading was taken.

@qty (float) Value of reading.

@meas_no (smallint) Unit of measure for value.

@comment_text (varchar 250) Comment associated with the reading.

Returned values:

None.

- Resources

- Processing
- Data

4.9.86 Ip_InsertSensorComment - Stored Procedure

- Purpose

Inserts a comment for a sensor.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

- Interface

Input Parameters:

@Sensor (int) The sensor number for the comment.

@Comment_Text (varchar 250) The text of the comment.

@Comment_DTTM (datetime) The date-time for the comment.

Returned values:

None.

- Resources
- Processing
- Data

4.9.87 Ip_InsertTPTR - Stored Procedure

- Purpose

Inserts a reading in the remperature readings table. It relies on the triggers and defaults in the TPTR_READING table to add certain values to the row

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

TPTR_READING

- Interface

Input Parameters:

@rcsys_no (smallint) Reading collection system for the reading.

@snsr_no (int) Sensor for the reading.

@dtm (datetime) Date-time for the reading.

@nstructr_id (char 10) Tank where the reading was taken.

@meas_no (smallint) Unit of measure for the reading.

@mrdcnam_no (smallint) COMmon name of the device used to take the reading.

@qty (float) Value for the reading.

@comment_text (varchar 250) Any comment associated with the reading.

Returned values:

None.

- Resources
- Processing
- Data

4.9.88 Ip_Is_MRDCNAM_NAME_Valid - Stored Procedure

- Purpose

Will return the Manual Reading Device Common Name Number that is associated with the given common name. If the common name is not in the table then this procedure returns no rows.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

MAN_RDG_DEV_COMN_NAME

- Interface

Input Parameters:

@mrdcnam_name (char 20) Common name for reading device.

Returned values:

Manual reading device common name number, reading device number.

- Resources

- Processing

- Data

4.9.89 Ip_LastSLVLReading - Stored Procedure

- Purpose

Returns the last surface level for a named structure.

- Function

Returns the last surface level reading for a named structure. The value is returned as ASCII characters, including the numeric value, in named columns so that this will work with the G2 product.

- Subordinates

None.

- Dependencies

Selects from Tables:

SLVL_READING

SENSOR

- Interface

Input Parameters:

@tankname (char 30) The tank to report on.

Returned values:

The character string for the last good surface level value, the suspect data status code, sensor type name, and date-time of the last good reading.

- Resources

- Processing

- Data

4.9.90 Ip_Load_User_SnsrPtsRetParm - Stored Procedure

- Purpose

Adds a new row to the U_SNSR_PTS_RET_PARM table each time it is called. The values are inserted in the table without range checking or anything.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

U_SNSR_PTS_RET_PARM

- Interface

Input Parameters:

@EMP_ID (char 7) Employee who owns the row.

@RCSYS_NO (smallint) Reading collection number for the row.

@SNSR_NO (int) Sensor number.

@BEG_DTTM (datetime) Beginning date time.

@END_DTTM (datetime) Ending date time.

@SNSR_TYPE_NAME (char 30) Type of sensor.

Returned values:

None.

- Resources

- Processing

- Data

4.9.91 Ip_LockRpt - Stored Procedure

- Purpose

Check to see if a user-defined lock exists for the LOCK_ID passed to it. Locks are stored in the tempdb..U_REPORT_LOCKS table if they exist. Ref. The Ip_UnlockRpt for removing this lock.

- Function

Checks to see if there is a lock on the table.

If not, the new employee is given the lock.

If so, and the employee ID is the same, it refreshes the lock and returns the employee ID.

If so, and the employee ID is not the same, and the lock is more than 1 hour old, the new employee is given the lock.

Otherwise, the lock is refused.

- Subordinates

None.

- Dependencies

Selects from Tables:

tempdb..U_REPORT_LOCKS

- Interface

Input Parameters:

@emp_id (EMP_ID) Employee who locked report.

@lock_id (smallint) ID of the things that is locked.

Returned values:

Employee ID of the person who owns the lock.

- Resources

- Processing

- Data

4.9.92 Ip_MakeFreqVioRpt - Stored Procedure

- Purpose

To select the data for the Frequency Violation Report.

- Function

Deletes the contents of the FREQ_VIO_REPORT table.

Selects the reading groups that have frequencies defined for them.

Calculates whether the has been too much time since the last good reading. (Suspect and transcribed readings are ignored.)

Marks the sample-type - fixed-sample-spaces according to the frequency they have violated.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
FREQ_VIO_REPORT
FSS_FREQUENCY
ILL_VALUE
RDG_COLCTN_SYS_SENSOR
RDG_COLCTN_SYS_SNSR_TYPE
READING_COLCTN_SYSTEM
SENSOR_TYPE
SLVL_READING
TPTR_READING

- Interface

Input Parameters:

None.

Returned values:

Values stored in the FREQ_VIO_REPORT table.

- Resources
- Processing
- Data

4.9.93 Ip_MakeTPTRCommentReport - Stored Procedure

- Purpose

Returns all the tank and temperature sensor comments for a tank for a given period of time.

- Function

Returns three result sets: temperature reading comments, sensor comments, and tank comments. Each one has slightly different contents as detailed below.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
NS_COMMENT
RDG_COLCTN_SYS_SENSOR
S_COMMENT
SENSOR
TPTR_READING

o Interface

Input Parameters:

- @tank (char 10) Tank to report on.
- @begin (datetime) Beginning date to report from.
- @end (datetime) Ending date time to report to.

Returned values:

Temperature Reading Comments:

Date-time, common name, collection-sensor index, reading comment.

Sensor Comments

Date-time, common name, collection-index, sensor comment.

Named Structure Comments

Date-time, tank name, NULL, named-structure comment.

o Resources

o Processing

o Data

4.9.94 Ip_ModifyNamedStructureComment - Stored Procedure

o Purpose

Updates a comment for a tank.

o Function

Same.

o Subordinates

None.

o Dependencies

Selects from Tables:

NS_COMMENT

o Interface

Input Parameters:

- @Comment_ID (numeric) The ID number for this comment.
- @Comment_Text (varchar 250) The text of the comment.

Returned values:

None.

o Resources

o Processing

o Data

4.9.95 Ip_ModifySensorComment - Stored Procedure

o Purpose

Modified the comment for a sensor.

o Function

Same.

o Subordinates

None.

o Dependencies

Selects from Tables:

S_COMMENT

o Interface

Input Parameters:

- @Comment_ID (numeric) The ID number for this comment.
- @Comment_Text (varchar 250) The text for the comment.

Returned values:

None.

- o Resources
- o Processing
- o Data

4.9.96 Ip_RemoveTankFromGroup - Stored Procedure

- o Purpose

Remove the given tank from the given tank group.

- o Function

Removes a tank from a given tank group by adding an end date-time to the relevant row.

- o Subordinates

None.

- o Dependencies

Selects from Tables:

TANK_IN_GROUP

- o Interface

Input Parameters:

@group_name (char 30) The tank group that this tank is no longer a member of.

@tank_name (char 10) The tank to modify.

@end_dttm (datetime) The end date-time for this relationship.

Returned values:

None.

- o Resources
- o Processing
- o Data

4.9.97 Ip_RetrieveData - Stored Procedure

- o Purpose

This procedure calls other procedures to retrieve data for each other data type.

- o Function

From the parameters decides which type of data are being selected by using the SENSOR_TYPE type to decode the type. Then it calls the appropriate other stored procedure to retrieve the data.

- o Subordinates

Ip_RetrieveSLVL

Ip_RetrieveILL

Ip_RetrieveTPTR

- o Dependencies.

Selects from Tables:

SENSOR_TYPE

- o Interface

Input Parameters:

@TankName (char 30) The tank we are working with.

@DttmBegin (datetime) Beginning date time to retrieve data.

@DttmEnd (datetime) Ending date time to retrieve data.

@GoodChosen (bit 1) Do we want good values? A non-zero value means yes.

@SuspectChosen (bit 1) Do we want suspect values? A non-zero value means yes.

@TChosen (bit 1) Do we want transcribed values? A non-zero value means yes.

@DataType (char 30) The type of data we are retrieving, e.g. 'ILL', 'Surface Level'.

@WindowSize (tinyint) The length of time over which to run averages and maximums

0 = daily

1 = weekly

2 = monthly

@AvgChosen (bit 1) Do we report average values? A non-zero value means yes.

@MaxChosen (bit 1) Do we report average values? A non-zero value means yes.

@AllData (bit 1) Do we report all data? A non-ser value means yes.

Returned values:

None directly. The stored procedures that it calls actually return the values.

- Resources
- Processing
- Data

4.9.98 Ip_RetrieveILL - Stored Procedure

- Purpose

Retrieve ILL data for the generalized data retrieval form.

- Function

Converts the passed selection flags into values that are used in the database.

Corrects the ILL values from feet to inches.

Prepares the appropriate headers for the data retrievals.

Retrieves the data requested in a format that the VB client understands.

- Subordinates

None.

- Dependencies

Selects from Tables:

ILL_VALUE

MEAS_CONVN_V

- Interface

Input Parameters:

@TankName (char 30) The tank we are working with.

@DttnBegin (datetime) Beginning date time to retrieve data.

@DttnEnd (datetime) Ending date time to retrieve data.

@GoodChosen (bit 1) Do we want good values? A non-zero value means yes.

@SuspectChosen (bit 1) Do we want suspect values? A non-zero value means yes.

@TChosen (bit 1) Do we want transcribed values? A non-zero value means yes.

@DataType (char 30) The type of data we are retrieving, e.g. 'ILL', 'Surface Level'.

@WindowSize (tinyint) The length of time over which to run averages and maximums

0 = daily

1 = weekly

2 = monthly

@AvgChosen (bit 1) Do we report average values? A non-zero value means yes.

@MaxChosen (bit 1) Do we report average values? A non-zero value means yes.

@AllData (bit 1) Do we report all data? A non-ser value means yes.

Returned values:

What is returned depends on what is selected. Values for one reporting frequency and any or all summary options may be returned. Each group of data returned is preceded by a header.

Header value, e.g. "MAXNTR||||| " for maximum neutron values.

Date. Format depends on the WindowSize parameter.

Daily: mm/dd/yy hh:mm

Weekly: yy week-number

Monthly: mm/yy

All data: mm/dd/yy hh:mm

Value, be it average, maximum, or all readings.

Suspect data status code.

- Resources
- Processing
- Data

4.9.99 Ip_RetrieveLLDataBatch - Stored Procedure

- ## Purpose

Selects all the ILL readings and compares them to the relevant baseline value for a specified time interval and puts them into the SURV_ALERTS_ILL table.

- ### Function

Select all the readings from the ILL_VALUE table for the specified date range and put them in a temporary table.

Update the readings with the number of days from today.

Update the readings units into inches.

Update the rows with the current baseline value.

Mark each row for the limits violated.

Insert the results into the SURV_ALERTS_ILL table.

- ## Subordinates

None.

- ## Dependencies

Selects from Tables:

SURV ALERTS ILL

ILL VALUE

MEAS CONVN_V

- ## Interface

Input Parameters:

@BegDate (datetime) Starting date to report on.

@EndDate (datetime) Ending date to report on.

Returned values:

None directly. Puts results in the SURV_ALERTS_ILL table.

- Resources
- Processing
- Data

4.9.100 Ip_RetrieveSLVL - Stored Procedure

- Purpose

Retrieve Surface Level data for the generalized data retrieval form.

- Function

Converts the passed selection flags into values that are used in the database.

Corrects the surface level values into inches if necessary.

Prepares the appropriate headers for the data retrievals. There are a bunch of these depending on the type of sampler and whether a summary or all records are chosen. Retrieves the data requested in a format that the VB client understands.

- Subordinates

None.

- Dependencies

Selects from Tables:

RDG_COLCTN_SYS_SENSOR

SENSOR

SLVL_READING

- Interface

Input Parameters:

@TankName (char 30) The tank we are working with.

@DtTmBegin (datetime) Beginning date time to retrieve data.

@DtTmEnd (datetime) Ending date time to retrieve data.

@GoodChosen (bit 1) Do we want good values? A non-zero value means yes.

@SuspectChosen (bit 1) Do we want suspect values? A non-zero value means yes.

@TChosen (bit 1) Do we want transcribed values? A non-zero value means yes.

@DataType (char 30) The type of data we are retrieving, e.g. 'ILL', 'Surface Level'.

@WindowSize (tinyint) The length of time over which to run averages and maximums

0 = daily

1 = weekly

2 = monthly

@AvgChosen (bit 1) Do we report average values? A non-zero value means yes.

@MaxChosen (bit 1) Do we report average values? A non-zero value means yes.

@AllData (bit 1) Do we report all data? A non-ser value means yes.

Returned values:

What is returned depends on what is selected. Values for one reporting frequency and any or all summary options may be returned. Each group of data returned is preceded by a header.

Header value, e.g. "MAXASL||" for maximum Auto FIC values.

Date. Format depends on the WindowSize parameter.

Daily: mm/dd/yy hh:mm

Weekly: yy week-number

Monthly: mm/yy

All data: mm/dd/yy hh:mm

Value, be it average, maximum, or all readings.

Suspect data status code.

- Resources

- Processing
- Data

4.9.101 Ip_RetrieveSLVLData - Stored Procedure

○ Purpose

Retrieves data from the SLVL_READING table for the reading modification form. The user may change the data retrieved.

○ Function

Retrieves the requested data. Returns it ordered by reading date-time.

• Subordinates

None.

○ Dependencies

Selects from Tables:

SENSOR
SLVL_READING

○ Interface

Input Parameters:

@rcsys_no (smallint 2) Reading collection system that we want.
@snsr_no (int) The specific sensor to retrieve.
@beg_dttm (datetime) The beginning date-time for readings.
@end_dttm (datetime) The ending date-time for readings.
@snsr_type (char 30) The type of sensor that we want.

Returned values:

Reading collection system number, sensor number, date (yy/mm/dd), time (hh:mm), named structure ID, surface level value, reading comment, surface level row ID.

- Resources
- Processing
- Data

4.9.102 Ip_RetrieveSLVLDataBatch - Stored Procedure

○ Purpose

Selects all the surface level readings for a specified time interval, compares them to the relevant baseline value and puts them into the SURV_ALERTS_SLVL table.

○ Function

Select all the readings from the SLVL_READING table for the specified date range and put them in a temporary table.

Update the readings with the number of days from today.

Update the readings units into the units used in the baseline.

Update the rows with the current baseline value.

Mark each row for the limits violated.

Insert the results into the SURV_ALERTS_SLVL table.

○ Subordinates

None.

○ Dependencies

Selects from Tables:

BASELINES
FIXED_SAMPLE_SPACE
RDG_COLCTN_SYS_SENSOR
READING_COLCTN_SYSTEM

SENSOR
 SENSOR_TYPE
 SLVL_READING
 SURV_ALERTS_SLVL

- Interface

Input Parameters:

 @BegDate (datetime) Beginning date-time to report on.

 @EndDate (datetime) Ending date-time to report on.

Returned values:

 None directly. Fill the SURV_ALERTS_SLVL table.

- Resources
- Processing
- Data

4.9.103 Ip_RetrieveTPTR - Stored Procedure

- Purpose

Retrieve temperature data for the generalized data retrieval form.

- Function

Converts the passed selection flags into values that are used in the database.

Corrects the temperature values from Celsius to Fahrenheit if necessary.

Prepares the appropriate headers for the data retrievals.

Retrieves the data requested in a format that the VB client understands.

- Subordinates

None.

- Dependencies

Selects from Tables:

 RDG_COLCTN_SYS_SENSOR

 SENSOR

 TPTR_READING

- Interface

Input Parameters:

 @TankName (char 30) The tank we are working with.

 @DttmBegin (datetime) Beginning date time to retrieve data.

 @DttmEnd (datetime) Ending date time to retrieve data.

 @GoodChosen (bit 1) Do we want good values? A non-zero value means yes.

 @SuspectChosen (bit 1) Do we want suspect values? A non-zero value means yes.

 @TChosen (bit 1) Do we want transcribed values? A non-zero value means yes.

 @DataType (char 30) The type of data we are retrieving, e.g. 'ILL', 'Surface Level'.

 @WindowSize (tinyint) The length of time over which to run averages and maximums

 0 = daily

 1 = weekly

 2 = monthly

 @AvgChosen (bit 1) Do we report average values? A non-zero value means yes.

 @MaxChosen (bit 1) Do we report average values? A non-zero value means yes.

 @AllData (bit 1) Do we report all data? A non-ser value means yes.

Returned values:

What is returned depends on what is selected. Values for one reporting frequency and any or all summary options may be returned. Each group of data returned is preceded by a header.

Header value, e.g. "AVGJT||||||||||||||||||||||||||||||||||||" for average J-type thermocouple values.

Date. Format depends on the WindowSize parameter.

Daily: mm/dd/yy hh:mm

Weekly: yy week-number

Monthly: mm/yy

All data: mm/dd/yy hh:mm

Value, be it average, maximum, or all readings.

Suspect data status code.

- Resources
- Processing
- Data

4.9.104 Ip_RetrieveTPTRDataBatch - Stored Procedure

- Purpose

Selects all the temperature readings for a specified time interval, compares them to the relevant baseline value and puts them into the SURV_ALERTS_TPTR table.

- Function

Select all the readings from the TPTR_READING table for the specified date range and put them in a temporary table.

Update the readings with the number of days from today.

Update the readings units into the units used in the baseline.

Update the rows with the current baseline value.

Mark each row for the limits violated.

Insert the results into the SURV_ALERTS_TPTR table.

- Subordinates

None.

- Dependencies

Selects from Tables:

BASELINES
FIXED_SAMPLE_SPACE
RDG_COLCTN_SYS_SENSOR
READING_COLCTN_SYSTEM
SENSOR
SENSOR_TYPE
SURV_ALERTS_TPTR
TPTR_READING

- Interface

Input Parameters:

@BegDate (datetime) Beginning date-time to report on.

@EndDate (datetime) Ending date-time to report on.

Returned values:

None directly. Fill the SURV_ALERTS_TPTR table.

- Resources
- Processing
- Data

4.9.105 Ip_RetrieveTPTR_byTC - Stored Procedure

Purpose

Retrieves temperature readings for a specific thermocouple (sensor/rcsys_no) for a date range. Used by the VB form when the user has hit the Retrieve Sensor Points button.

Function

Converts the passed selection flags into values that are used in the database.

Corrects the temperature values from Celsius to Fahrenheit if necessary.

Prepares the appropriate headers for the data retrievals.

Retrieves the data requested in a format that the VB client understands.

- **Subordinates**

None.

Dependencies

Selects from Tables:

RDG_COLCTN_SYS_SENSOR

SENSOR

TPTR_READING

Interface

Input Parameters:

@DttmBegin (datetime) Beginning date time to retrieve data.

@DttmEnd (datetime) Ending date time to retrieve data.

@GoodChosen (bit 1) Do we want good values? A non-zero value means yes.

@SuspectChosen (bit 1) Do we want suspect values? A non-zero value means yes.

@TChosen (bit 1) Do we want transcribed values? A non-zero value means yes.

@DataType (char 30) The type of data we are retrieving, e.g. 'ILL', 'Surface Level'.

@rcsys_no (smallint) Reading collection system to report on.

@snsr_no (int) Sensor to report on.

@WindowSize (tinyint) The length of time over which to run averages and maximums

0 = daily

1 = weekly

2 = monthly

@AvgChosen (bit 1) Do we report average values? A non-zero value means yes.

@MaxChosen (bit 1) Do we report average values? A non-zero value means yes.

@AllData (bit 1) Do we report all data? A non-ser value means yes.

Returned values:

[illegible]

Date. Format depends on the WindowSize parameter.

Daily: mm/dd/yy hh:mm

Weekly: yy week-number

Monthly: mm/yy

All data: mm/dd/yy hh:mm

Value, be it average, maximum, or all readings.

Suspect data status code.

Resources

Processing

Data

4.9.106 Ip_RetrieveVVBaselines - Stored Procedure

- Purpose

Return the baselines and tolerances for a given tank and data type.

- Function

Returns the baseline value, start date, slope rate, and limits. For temperatures it returns the value for only 1 sensor. For surface levels and ILLs it returns the values for the tank.

- Subordinates

None.

- Dependencies

Selects from Tables:

BASELINES

SENSOR_TYPE

- Interface

Input Parameters:

@tank (char 10) Tank to report on.

@data_type (char 30) Type of sensor to report on.

@snsr_no (int) Sensor number to report on.

Returned values:

Baseline value, Start date-time, baseline date-time, slope rate, relative upper limit, relative lower limit. For temperatures it returns only the baseline for one sensor; otherwise it returns the value for the type of sensor in the tank.

- Resources

- Processing

- Data

4.9.107 Ip_TemperatureReport - Stored Procedure

- Purpose

Populates the table U_TEMPERATURE_REPORTS for all the tanks in a tank group. This table is used by Max Temperature Report for reporting. Only one copy of this proc should be run at a time since it empties and fills the U_TEMPERATURE_REPORTS table.

- Function

Empties the U_TEMPERATURE_REPORTS table.

Gets a list of all tanks in the given group.

Finds maximum and minimum surface levels for the date range for the tanks.

Gets a list of all temperature sensors for all requested tanks.

Gets all the readings for the temperature sensors in the date range.

Decides if the reading is in the waste or not using the SLVL max and min values. (This greatly reduces the number of calculations that you have to do.)

For TC sensors that are between the max and min SLVL readings, decides if these are in the waste or not.

If a surface level can't be established for the reading that reading is reported on both the IN and OUT of waste reports.

Depending on the in-out flag, Insert the appropriate readings into the U_TEMPERATURE_REPORTS table.

- Subordinates

None.

- Dependencies

Selects from Tables:

FIXED_SAMPLE_SPACE
 FSS_SENSOR
 RDG_COLCTN_SYS_SENSOR
 SENSOR
 SENSOR_TYPE
 SLVL_READING
 TANK_IN_GROUP
 U_TEMPERATURE_REPORTS

Interface

Input Parameters:

@tank_group (char 30) Tank group to report on.
 @start_date (datetime) Starting date-time to get readings.
 @end_date (datetime) Ending date-time to get readings.
 @in_out (char 3) Report on readings in or out of the waste? "IN" = in,
 "OUT"=out.

Returned values:

None directly. Fill the U_TEMPERATURE_REPORTS table.

- Resources
- Processing
- Data

4.9.108 Ip_UnlockRpt - Stored Procedure

Purpose

Removes the user-defined lock established by Ip_LockRpt.

Function

Removes the line from the tempdb..U_REPORT_LOCKS table for the passed-in lock value.

Subordinates

None.

Dependencies

Selects from Tables:

tempdb..U_REPORT_LOCKS

Interface

Input Parameters:

@lock_id (tinyint) The lock ID to remove.

Returned values:

None.

- Resources
- Processing
- Data

4.9.109 Ip_UpdateBaseline - Stored Procedure

Purpose

Updates a record in the BASELINES table.

Function

Converts the measure names to our internal codes.

Update the baseline record.

RAISERROR if there is a problem.

Subordinates

None.

Dependencies

Selects from Tables:

BASELINES

MEASURE

• **Interface****Input Parameters:**

- @TankName (char 10) Name of tank where baseline is located.
- @SnsrTypeName (char 30) Type of sensor to update baseline for.
- @SnsrNo (int) Sensor number.
- @StartDttm (datetime) Starting date-time for baseline.
- @BaselineValue (real) Value of baseline at start date.
- @BaselineDttm (datetime) Baseline date-time.
- @BsinMeasName (char 40) Unit of measure for the baseline.
- @SlopeRate (real) Value of the slope rate.
- @SlopeMeasName (char 40) Unit of measure for the slope rate.
- @RelUpLmt (real) Relative upper limit value.
- @RelLowLmt (real) Relative lower limit value.
- @Empld (char 7) Employee who made the change.

Returned values:

None.

- Resources
- Processing
- Data

4.9.110 Ip_UpdateSLVL - Stored Procedure• **Purpose**

Given a SLVL_ID, which points to a single row in the SLVL_READING table, updates the record with the passed parameters.

• **Function**

Same.

• **Subordinates**

None.

• **Dependencies****Selects from Tables:**

SLVL_READING

• **Interface****Input Parameters:**

- @new_rcsys_no (smallint) Reading collection system number to update.
- @new_snsr_no (int) Sensor number to update.
- @new_dttm (datetime) Date-time of reading to update.
- @new_nstructr_id (char 10) Name of named structure to update.
- @new_qty (float) Quantity to update.
- @new_comment_text (varchar 250) Comment to update.
- @slvl_id (numeric) The SLVL_ID of the row to update.

Returned values:

None.

- Resources
- Processing
- Data

4.9.111 Ip_UpdateTPTR - Stored Procedure

- Purpose

Given a TPTR_ID, which points to a single row in the TPTR_READING table, updates the record with the passed parameters.

- Function

Same.

- Subordinates

None.

- Dependencies

Selects from Tables:

TPTR_READING

- Interface

Input Parameters:

@new_dttm (datetime) Date-time of reading to update.

@new_qty (float) Quantity to update.

@new_mrdcnam_no (smallint) manual reading device common name to update.

@new_meas_name (char 40) measure name to update.

@comment_text (varchar 250) Comment values to update.

@tptr_id (numeric) The TPTR_ID of the row to update.

Returned values:

None.

- Resources

- Processing

- Data

4.9.112 Ip_Updt_SuspDataStatCode - Stored Procedure

- Purpose

Updates the SUSP_DATA_STAT_CODE for a given type of sensor with a reading at a given time.

- Function

Decides which table to update.

Update the SUSP_DATA_STAT_CODE to the passed value in the relevant table.

- Subordinates

None.

- Dependencies

Selects from Tables:

ILL_VALUE

SLVL_READING

TPTR_READING

- Interface

Input Parameters:

@ReadingType (char 15) Type of reading update, e.g. "Temperature", "Surface Level", or "ILL".

@Dttm (datetime) Date-time of reading to update.

@RcSysno (smallint) Reading collection system number for reading.

@SnsrNo (int) Sensor to update.

@SuspDataStatCode (char 1) The new value for the SUSP_DATA_STAT_CODE.

@Empld (char 7) The employee who is making the changes.

Returned values:

None.

- Resources
- Processing
- Data

Index

Chart Builder, 86

Crystal Reports, 28, 29, 30, 45, 86

Graphics Server, 86

HDAT, 4, 8, 10, 31, 68

TMACS, 2, 4, 9, 12, 17, 18, 19, 23, 24, 31,
53, 68, 75, 94, 95, 96

TWINS, 4, 9, 13, 30

DISTRIBUTION SHEET

To Distribution	From Waste Tank Surveillance Engineering, and Technical Applications Resources	Page 1 of 1
		Date 10/02/95
Project Title/Work Order Surveillance Analysis Computer System (SACS) Software Design Document (SDD) Rev. 1		EDT No.
		ECN No. 626411

Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
M. J. Holm	R1-01	X			
L. M. Ollero	R1-01	X			
M. J. Winkelman	S3-30	X			
Central Files	A3-88	X			