CTBTO preparatory commission for the comprehensive nuclear-test-ban treaty organization
PREPARATORY COMMISSION

IDC/DEVLAN/OG-RN
20 May 2008

English only

AWST science software technology

AWST-TR-08/06

IDC Development LAN Operator's Guide - RN

This document contains information on the operation of the Radionuclide Software on the IDC Development LAN, including baseline descriptions and monitoring, troubleshooting and change procedures.

Summary

This document describes the procedures used to operate and maintain the Radionuclide (RN) processing software on the IDC Development LAN. It also describes the hardware, software and environment baseline, which is needed to properly operate the Development LAN software. The chapters on the Development LAN baseline and on change procedures are of general interest for all Development LAN users and developers.

Document history

Version	Date	Author	Description
0.1	7 June 2004	Alexander Boresch	draft document structure
0.2	12 August 2004	Marian Harustak	partial draft procedures in chapters 3 and 4
0.3	16 September 2004	Alexander Boresch	complete draft version
0.4	15 October 2004	Alexander Boresch	revised draft version
1.0	28 November 2004	Alexander Boresch	final SHI version
1.1	16 August 2005	Alexander Boresch	baseline for expanded version (SHI, RN, ATM)
1.2	7 February 2006	Henrik Edlund	ATM chapters added
1.3	14 December 2006	Gertrud Waich	added autoconf build procedures, updated SHI chapters
2.0	5 January 2007	Gertrud Waich, Alexander Boresch	full SHI, RN and ATM version
3.0	20 May 2008	Vladimir Gelashvili, Gertrud Waich, Jan Wüster (review)	document re-structured, updated and split into separate volumes for SHI, RN and ATM software

Contents

List of Figures

1. INTRODUCTION

1.1. Identification and Purpose of the Document

This document describes the procedures used to operate and maintain the radionuclide (RN) processing software on the IDC Development LAN. It provides an overview over the existing infrastructure and also describes the hardware and software environments, which is needed to properly operate the Development LAN software. The addressed audience is primarily the Development LAN operators, but also includes all developers, testers, users and administrators of the Development LAN hardware and software.

There are two companion documents, [IDC/DEVLAN/OG-SHI] and [IDC/DEVLAN/OG-ATM], which describe the Development LAN's Operations and Maintenance procedures for the waveform technologies processing software and the Atmospheric Transport modelling software, respectively.

The current document is based on general framework provided by the following documents:

- IDC Operational Manual, [IDC-OM]: Describes the general operational framework and mission of the IDC. Available at
- H:\Conference Documents\Official Documents\PMOs and AG\WGA\WGB\Operational Manuals\IDC OM in the CTBTO Intranet.

- Draft Procedures of the IDC Configuration Control Board (CCB), [IDC-CCB]: Describes the software and configuration change procedures for the IDC Testbed and Operations LANs. Available at: http://idc030.idc.ctbto.org:8000/Docs/CCB-Documentation.html in the CTBTO Intranet.

- IDC Software Documentation Orientation Booklet, [IDC-SDO]: Introduces new users to the IDC work environment including the IDC computer infrastructure and software. Available at: http://intranet.ctbto.org/stafforientation02mar20.pdf.

1.2. Purpose and Role of the Development LAN

The Development LAN was created in July 2002 and serves as a development and test platform for the IDC software. This includes processing software for all CTBT monitoring technologies Seismic, Hydro-acoustic and Infrasound (SHI) data processing, Radionuclide (RN) data processing and Atmospheric Transport Modelling (ATM).

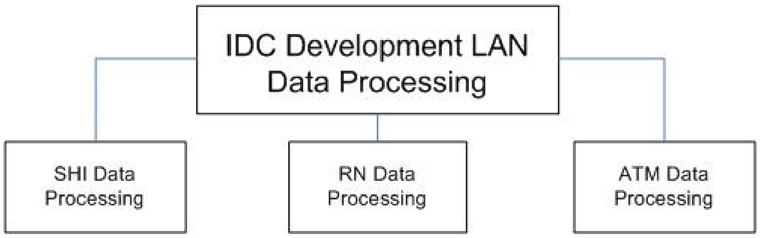


Figure 1 IDC technologies and data processing systems

The Development LAN is owned by the IDC Software Integration unit and is under less rigorous configuration control than other IDC LANs (Testbed and Operations LANs). When new or modified software becomes available and has successfully passed unit testing by the developers, it is installed on the Development LAN to be tested in the integrated IDC data processing system for the relevant technology. Such integration testing checks if the new or modified software component functions well within the framework of the whole SHI processing system and helps to identify any unintended effects of the installed software change on other processing software. If such effects are found, they are analyzed and resolved before the software change is implemented on the Testbed LAN for operational testing, and finally on the Operations LAN for use in regular IDC operations. Thus the Development LAN has a major quality assurance role in the IDC software development cycle.

1.3. Document Overview

Chapter 1 gives an introduction to this document.

Chapter 2 contains general information, which is common for all processing technologies. It is organized in sections describing roles and responsibilities, the hardware and infrastructure baselines, the general directory structure, the role and use of the configuration management system *ClearCase* at the IDC, and generic change procedures. The information in this chapter is of general interest for all Development LAN users. **Chapter 2 is identical for all three volumes of the Development LAN Operator's Guide. To support maintainability it is only contained in [DEVLAN/OG-SHI] and omitted here.**

Chapter 3 contains all RN-specific information including the baseline inventory and procedures for routine operations, system monitoring, system maintenance and troubleshooting. The primary audience for this chapter are the operators of the RN processing software. However, the baseline information and troubleshooting procedures may also be of interest for developers and testers.

1.4. Typographical Conventions

Element	Font	Example
application, process, programming language	*italic*	*WorkFlow*, *Perl*
database account, database table, database attribute	UPPERCASE.**bold**.normal	IDCX.**interval**.intvlid

configuration parameter, variable, literal value, command, computer code, machine name	`Courier New 11`	`ssh cmss@eldey,`
		`$(CMS_CONFIG)`

2. GENERAL BASELINES AND PROCEDURES FOR ALL TECHNOLOGIES

This chapter is identical to the three volumes of the Development LAN Operator's Guide. To facilitate document maintenance it is omitted here. See [IDC/DEVLAN/OG-SHI] to see the contents.

3. RADIONUCLIDE PROCESSING

3.1. Introduction

Radionuclide stations sample the atmosphere in time intervals of 24 hours or less with the objective of detecting traces of radioactive particles, gases and aerosols that may have originated from a nuclear weapon test.

- Atmospheric, underwater, surface and near surface (non-evasive) explosions can be identified via particulate radionuclide monitoring. Nearly all fission and activation products created in a nuclear explosion are solid elements at ambient temperatures. They therefore attach to particles, which either exist in the form of natural aerosols or are formed by solidified residues of the explosive device or its environment. The filtering devices used in the particulate station network collect these radioactive particles.
- With underground and underwater (evasive) explosions, there is little probability that the solid residues will enter the atmosphere and be available for collection on the filtering devices. In these cases no aerosols, but only gases are released into the atmosphere via cracks, diffusion, etc. Of special interest are gases, which have little affection to surfaces of underground cracks and fissures, nor will dissolve appreciably in water. This applies especially to the noble gases helium, neon, argon, krypton, xenon and radon. The only radioactive isotopes of these elements expected after a nuclear detonation are isotopes of argon, krypton and xenon. Collection of argon is complex and therefore too expensive. Almost all krypton isotopes are too short-lived for collection; one is too long lived, and therefore has a high background in the atmosphere from all kinds of utilization of fission. Radon isotopes are not present in nuclear weapons debris in quantities that can compete with the ambient background stemming from primordial thorium and uranium. Just several isotopes of xenon have suitable half-lives and radiations to be detected. The CTBT relevant radioxenon isotopes are: ^{131m}Xe, ^{133m}Xe, ^{133}Xe, ^{135}Xe with half-lives of 11.934, 2.19 and 5.243 days, and 9.14h, respectively. These radioactive xenon isotopes are also produced in nuclear power plants and, directly or as by-product, in certain medical facilities. To distinguish between the release from such applications and from a subsurface nuclear explosion, the ratios of the isotopes are helpful.

Particulate stations are required to report sample data to the IDC within 72 hours after the commencement of sample collection. During this time period, particulate samples are allowed a substantial decay time and then counted on a High-Purity Germanium (HPGe) detector for at least 20 hours.

Noble gas stations have different requirements for processing times. These stations are required to report sample data to the IDC within 48 hours after the commencement of sample collection. Noble gas samples are chemically processed after sample collection and then counted for a maximum of 24 hours. Acceptable measurement modes for noble gas samples include high resolution gamma (γ) spectroscopy and beta-gamma (β-γ) coincidence. Chemical processing entails passing the noble gas sample through various traps and sieves to separate xenon (Xe) from other atmospheric gases that can interfere with the detection of radiation emitted by the radioxenons. Currently there are four radioxenon measurement systems in operational use:

ARSA (Automated Radioxenon Sampler and Analyzer)
Developer: PNNL (USA)
Detection: 2-D beta gamma
Technology: coincidence spectroscopy
Collection Time: 8 hours per cycle

SAUNA (Swedish Automatic Unit for Noble gas Acquisition)
Developer: FOI (Sweden)
Detection: 2-D beta gamma
Technology: coincidence spectroscopy
Collection Time: 12 hours per cycle

ARIX (Analyzer of Radioactive Isotopes of Xenon)
Developer: KRI (Russian Federation)
Detection: beta-gated gamma
Technology: spectroscopy
Collection Time: 8 hours per cycle

SPALAX (Système de Prélèvement Automatique en Ligne avec l'Analyse du Xénon)
Developer: CEA (France)
Detection: high resolution gamma
Technology: spectroscopy
Collection Time: 24 hours per cycle

The objective of radionuclide processing is to analyze radionuclide data and produce standard radionuclide data products. To achieve these objectives all relevant radionuclide data received from the operating IMS radionuclide stations must be successfully processed. There are several types of the IMS radionuclide data:

- Pulse Height Data (PHD), containing the spectra to be analyzed automatically and interactively
- Meteorological (MET) Data
- State-Of-Health (SOH) Data of a station and its associated detector
- Alert Data
- Data from Certified Radionuclide Laboratories (RLR)

PHD messages are automatically analyzed. The following types of PHD messages are recognized:

- BLANKPHD messages contain PHD acquired by counting an unexposed air filter on an HPGe detector. BLANKPHD messages are not sent from noble gas stations.
- CALIBPHD messages contain PHD acquired by counting a known standard source with a detector system.
- DETBKPHD messages contain PHD acquired by performing a background measurement of a detector system.
- GASBKPHD messages contain PHD of an empty plastic scintillation gas cell from stations that observe a memory effect during sample acquisition due to nuclides from the previous sample absorbed onto the walls of the gas cell. At present, only noble gas monitoring systems that utilize beta-gamma coincidence counting have plastic scintillation gas cells. The GASBKPHD is acquired after a sample has been evacuated from the gas cell and before the next sample acquisition is started. The purpose of the GASBKPHD is to enable the quantification of radioxenon atoms that are adsorbed onto the walls of the plastic scintillation gas cell.
- QCPHD messages contain the PHD of the daily quality control measurement acquired from a brief count of a known standard source with a detector system
- SPHD messages can contain the actual Sample Pulse Height Data, simply called “Samples” in the jargon. Samples are further subdivided into:
    - SPHDF: Full sample, contains PHD from a sample acquired for the IDC-defined full acquisition time
    - SPHDP: Preliminary sample, contains PHD from a sample acquired for a time shorter than that of the full acquisition time.

The automatic processing also serves to aid in monitoring the state-of-health of a station and its associated detector.

3.2. Baseline Inventory

3.2.1. Hardware baseline

The automatic RN processing software is operated on a single machine (`eldey`).

The RN software uses the standard Development LAN database server `orac1` and the database instances `idcdev` as a main database and `idcdev2` as an archive database.

Interactive RN analysis is not performed routinely on the Development LAN but can be configured on any Development LAN machines when desired.

3.2.2. Application software baseline

The automatic RN processing software is run by the *UNIX/Linux* user `auto`. The user `auto` is maintained by the Development LAN operator.

The interactive RN analysis software is run by individual analysts from their UNIX accounts, and by other users who wish to perform interactive analysis of RN data. On the Development LAN interactive RN analysis is not routinely performed.

Several RN software applications or subsystems have been replaced in the recent years or are planned to be replaced in the future. The existing software design and user manuals from the original IDC documentation set are partially obsolete. The two most relevant documents are [IDC5.2.2] IDC Processing of Radionuclide Data, Revision 3, and [IDC6.5.10] Radionuclide Software User Manual, Revision 4. They are electronically available at the IDC products and services web site: http://kuredu.ops.ctbto.org.
A revision of the documentation, parallel to software redesign and development, is in progress.

3.2.3. Software location

All RN software applications are located in the directory `/dvl/software/rn`. Executable files are installed in the `s9rn` subdirectory,

All RN scripts that are part of the application baseline are located in the directory `/dvl/software/rn/scripts`. The scripting languages used for baseline scripts are *sh, csh* and *Perl*

All software modules (source code and executables binaries) and script versions are version-controlled using the IDC's configuration management system *ClearCase*. The Development LAN Operator maintains the application software and scripts baseline in *ClearCase* and in the Development LAN runtime system. The *ClearCase* administrator maintains the IDC software baseline for other LANs and assigns the Development LAN software patch labels. Refer to section 1.10 of the SHI part of the Operator's Guide for additional information on requesting and implementing changes to the Development LAN application software.

Source code can be found in the `code` directory and its subdirectories. The individual applications are grouped into subdirectories of the `code/rms` directory. The most relevant subdirectories are `automatic`, `interactive`, `java`, `libraries` and `utilities`. See section 3.2.5. for more information on the RN directory structure.

3.2.4. Software configuration baseline

All configuration files for the RN processing system are located in the directory `/dvl/software/rn/configs`. The `configs` directory has subdirectories for environment configuration files (`envs`), parameterfiles (`parmfiles`), resource files for *Inspectra* and *WorkFlow* and icon pixmaps used by the *X Window System* for these applications (`resources`), *SQL* queries used at various steps in the processing chain and by *WorkFlow* and a file with anthropogenic nuclide names (`NUCLIDE`).

The environment files in the `envs` directory contain development related environments (`rms_dev.env`) and settings for analysts (`rms_user.env`). The *WorkFlow* configuration parameters are stored in the `parmfiles` directory.

3.2.5. Development LAN RN directory structure

The general top-level directory structure for the Development LAN is described in chapter 2. The RN-specific top level directories follow this general structure and are listed in the table below. Directory levels are counted from the top level (level 1), which is `$(lan)=/dvl` for the Development LAN. In the RN area, the subdirectory depth goes to 8.

3.2.5.1. *Top level RN directories*

Top level (Level 3) RN directories	
`$(lan)/data/rn`	data directories and files used by the RN software
`$(lan)/logs/rn`	RN log file area
`$(lan)/products/rn`	RN data products
`$(lan)/software/rn`	installed RN software (source code, binaries, libraries, scripts, configuration files)

3.2.5.2. `software` *directory*

The RN software is structured into directories according to the type of the files (binaries, libraries, source code, etc.).

Level 4: RN software	
`$(lan)/software/rn`	RN software directory
`./MigrateData`	data migration software
`./RNArchive`	obsolete
`./RNMigrateData`	obsolete
`./code`	a copy of the entire source code structure for automatic and interactive applications, libraries, utilities, and documentation (All source code is maintained and built in the *ClearCase* VOB `/vobs/idc/rms`)
`./configs`	RN configuration directory tree (including RN parameter files)
`./genie`	external vendor library (Canberra software)
`./java`	*Java* classes and images
`./man`	man pages for RN software
`./s9rn`	platform specific binary and library directory
`./saint`	software for the Simulation Assisted Interactive Nuclide-review Tool
`./scripts`	platform-independent scripts
`./sparc-SunOS5.6`	symbolic link to `s9rn`
`./sparc-SunOS5.6.retired`	obsolete
`./sparc-SunOS5.7`	symbolic link to `sparc-SunOS5.6`
`./sparc-SunOS5.8`	symbolic link to `sparc-SunOS5.6`
`./sparc-SunOS5.9`	symbolic link to `sparc-SunOS5.6`

3.2.5.3. `MigrateData` *directory*

Level 5: Data migration (archival) software directory	
`$(lan)/software/rn/MigrateData`	RN data migration software
`./archdb`	database maintenance SQL code, schema and documentation for the archival database
`./bin`	binaries
`./bothdb`	database maintenance SQL code, schema and documentation for operational and archival databases
`./cron`	*crontab* files
`./opsdb`	database maintenance SQL code, schema and documentation for the operational database
`./rmsmigrate`	parameter files with configuration rules for migration
`./workflow`	parameter files for *WorkFlow* monitoring of migration

3.2.5.4. `code` *directory*

Level 5: Source code directory	
`$(lan)/software/rn/code`	RN software source code copy from *ClearCase* VOB
`./build`	contains *Makefile*s and templates for building a new release
`./rms`	storage location for source code of all RN applications
`./target`	compiled Solaris binaries
`./web`	web subsystem files

3.2.5.5. `configs` *directory*

Level 5: Configuration directory	
`$(lan)/software/rn/configs`	RN software configuration
`./envs`	Default user environment (shell configuration and forward files) for `rmsops` user, analysts and developers.
`./old_envs`	obsolete
`./parmfiles`	parameter files for *WorkFlow* monitoring of automatic processing
`./resources`	*X* resource files for *WorkFlow* and *Inspectra*, and XPM images for *Inspectra*
`./sql`	SQL queries used at various steps in the processing chain and by *WorkFlow*

3.2.5.6. `genie` *directory*

External vendor library from Canberra.

3.2.5.7. `java` *directory*

Level 5: Java directory	
`$(lan)/software/rn/java`	RN *Java* libraries
`./classes`	*Java* classes
`./images`	images used by *Java* applications

3.2.5.8. `man` *directory*

Level 5: Manual page directory	
`$(lan)/software/rn/man`	RN man pages
`./man1`	man pages for software applications

3.2.5.9. `s9rn` *directory*

Level 5: Solaris 9 runtime directory	
`$(lan)/software/rn/s9rn`	RN Solaris 9 runtime binaries and libraries
`./bin`	binaries
`./lib`	libraries
`./oraform`	Oracle forms

3.2.5.10. `data` *directory*

All RN data are located in `$(lan)/data/rn`. There are several subdirectories for different types of data and messages.

Level 4: RN data directories	
`$(lan)/data/rn`	RN data directory
`./atm`	unused
`./cnf`	unused
`./histogram`	histogram data
`./mail`	sent emails
`./new`	unused
`./other`	unused
`./processed`	processed data (messages)
`./spectrum`	spectrum data

3.2.5.11. `processed` *directory*

A copy of processed messages is stored in subdirectories depending on the message type.

Level 5: RN processed messages directories	
`$(lan)/data/rn/processed`	RN processed messages directory
`./alert`	ALERT data messages
`./blank`	BLANKPHD messages
`./cal`	CALIBPHD messages
`./detbk`	DETBKPHD messages
`./gasbk`	GASBKPHD messages
`./met`	MET data file messages
`./qc`	QCPHD messages
`./rlr`	Radionuclide Laboratory Report; contains sample analysis results from a certified radionuclide laboratory.
`./sample`	SPHD messages
`./soh`	SOH data messages

3.2.5.12. `logs` *directory*

All RN software log files are located in `$(lan)/logs/rn`.

Level 4: RN logs directories	
`$(lan)/logs/rn`	RN logs directory
`./CINs`	logs from older Change Implementation Notes. Currently obsolete.
`./DBflags`	logs from *rms_DB_flags* (when called from *rms_process*)
`./MigrateData`	logs from data migration
`./analyze`	logs from *rms_analyze*
`./arix`	unused
`./categorize`	logs from *rms_categorize* for particulate samples
`./inspectra`	logs from *rms_inspectra*
`./mar`	logs from *rms_mar_auto*
`./pipeline`	logs from *rms_pipeline*, *rms_input* and *rms_process*
`./qat`	unused
`./rnps`	unused
`./xanalyze`	logs from *rms_xanalyze* and *bg_analyze*
`./xcategorize`	logs from *rms_xcategorize* for noble gas samples

3.2.5.13. `products` *directory*

All RN data products are located in `$(lan)/products/rn`.

Level 4: RN products directories	
`$(lan)/products/rn`	RN products directory
`./arr`	Automatic Radionuclide Reports
`./rlr`	Radionuclide Laboratory Reports
`./rnps`	unused
`./rrr`	Reviewed Radionuclide Reports. A revised version of the ARR, including the results of interactive review of a radionuclide sample.
`./ssreb`	Standard Screened Radionuclide Event Bulletins
`./temp_rrr`	obsolete
`./temp_ssreb`	obsolete

3.2.6. Development LAN RN user environment

3.2.6.1. *Automatic processing user environment*

All RN automatic processing is an extension of the SHO Message Subsystem: The Message Subsystem application *ParseData* calls the RN script *rms_pipeline* and this in turn calls all the other components of the radionuclide automatic processing pipeline. Therefore the user environment for all RN automatic applications is the environment of the technical user running the Message Subsystem, i.e. the environment of `auto` on `eldey`.

3.2.6.2. *Interactive software user environment*

The user environment settings are defined in the file `.cshrc` in the home directory of the RN users and therefore automatically sourced at login time, except for the account `rmscm.`

The `rmscm` login is used to build all RN software on the Development LAN. Before building any software the file `release_build.cshrc` has to be sourced.

In order to allow interactive users to use the *inspectra* and RN applications, proper environment settings relevant to MATLAB have to be set. The file `rms_user.env` has to be sourced and `MATLAB_HOME` has to be set `to /opt/Matlab` to block out the 'normal'system MATLAB6.

3.2.7. RN data acquisition and processing in the Development LAN – overview

The RN software is generally operated independently of the software for other technologies. However, it depends on the SHI message subsystem for data acquisition and on the web subsystem to make data products available on the IDC website. Refer to chapter 2 for general roles and responsibilities on the Development LAN. The following RN-specific UNIX users and *Oracle* database accounts are defined:

`rmsops` This account exists to hold links and common files used by the RMS processing pipeline and all RMS analysts. The user does not run any software, nor does it have any other specific functions. In the home directory of this user no other information than these common files and links should be stored.

`rmscm` This account is used to build all RN software on the Development LAN.

`rmsauto` *Oracle* database account; used for automatic analysis of sample messages as they arrive

`rmsman` *Oracle* database account; used for interactive analysis by the IDC radionuclide analysts

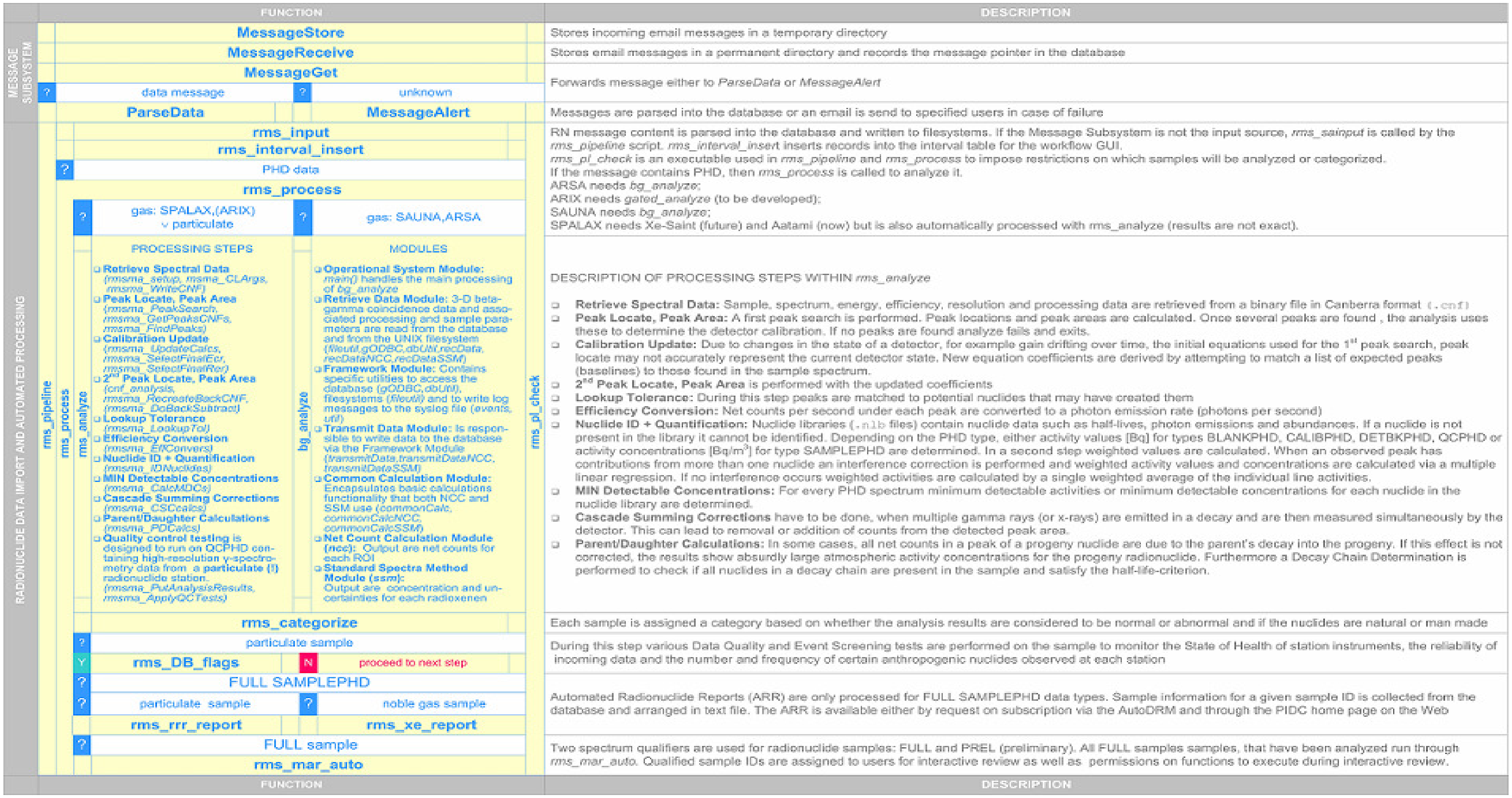
A two-tiered database is set up for analysis of IMS radionuclide data. One database account (`rmsauto`) is used for automatic analysis of sample data as it arrives. The second database account (`rmsman`) is used for interactive analysis by the IDC radionuclide analysts. This two-tiered database schema is designed to preserve both the automatic and interactive analysis results without overwriting each other. The automatic RN processing pipeline exclusively connects to the `rmsauto` database account, while RN analysts always work with the `rmsman` account. Five database tables residing outside the RN database schema are relevant for radionuclide processing. They are IDCX**.lastid**, IDCX**.fileproduct**, IDCX**.fpdescription**, IDCX**.msgdisc** and STATIC**.affiliation** where the IDCX and STATIC database accounts are owned by the SHI software. These tables are interfaces to the SHI software and are used to identify incoming data messages (**msgdisc**), to make RN data products available to the SHI subscription subsystem (**fpdescription**, **fileproduct**) and to track unique identifier numbers for entries in these tables (**lastid**).
The radionuclide database tables are conceptually organized into four groups:

- Raw Data Tables: Contain original data as received from IMS stations. Owner: `IDCX`
- Static Data Tables: Contain configuration data for detectors, stations and tables used as default processing parameters for sample data analysis. Owner: `rmsman`
- Analysis Data Tables: Contain the analysis results of data processing. Analysis data is first inserted into `rmsauto` and then moved to the `rmsman` account by database triggers. Owner: `rmsauto` and `rmsman`.
- Independent Data Tables: Contain the results of tests and analysis performed on data in other tables of the current account. These tables exist independently in both `rmsauto` and `rmsman` accounts.

For a detailed description of all 175 radionuclide database tables please see APPENDIX A. Description of Radionuclide Database Tables

3.2.8. RN automatic processing in the Development LAN – overview

The table in Figure 2 shows the complete RN pipeline



PROCESSING STEPS

- ❑ **Retrieve Spectral Data** *(rmsma_setup, rmsma_CLArgs, rmsma_WriteCNF)*
- ❑ **Peak Locate, Peak Area** *(rmsma_PeakSearch, rmsma_GetPeaksCNFs, rmsma_FindPeaks)*
- ❑ **Calibration Update** *(rmsma_UpdateCalcs, rmsma_SelectFinalEcr, rmsma_SelectFinalRer)*
- ❑ **2nd Peak Locate, Peak Area** *(cnf_analysis, rmsma_RecreateBackCNF, (rmsma_DoBackSubtract)*
- ❑ **Lookup Tolerance** *(rmsma_LookupTol)*
- ❑ **Efficiency Conversion** *(rmsma_EffConvers)*
- ❑ **Nuclide ID + Quantification** *(rmsma_IDNuclides)*
- ❑ **MIN Detectable Concentrations** *(rmsma_CalcMDCs)*
- ❑ **Cascade Summing Corrections** *(rmsma_CSCcalcs)*
- ❑ **Parent/Daughter Calculations** *(rmsma_PDCalcs)*
- ❑ **Quality control testing** is designed to run on QCPHD containing high-resolution γ-spectrometry data from a **particulate (!)** radionuclide station. *(rmsma_PutAnalysisResults, rmsma_ApplyQCTests)*

MODULES

- ❑ **Operational System Module:** *main()* handles the main processing of *bg_analyze*
- ❑ **Retrieve Data Module:** 3-D beta-gamma coincidence data and associated processing and sample parameters are read from the database and from the UNIX filesystem *(fileutil,gODBC,dbUtil,recData, recDataNCC,recDataSSM)*
- ❑ **Framework Module:** Contains specific utilities to access the database *(gODBC,dbUtil)*, filesystems *(fileutil)* and to write log messages to the syslog file *(events, util)*
- ❑ **Transmit Data Module:** Is responsible to write data to the database via the Framework Module *(transmitData,transmitDataNCC, transmitDataSSM)*
- ❑ **Common Calculation Module:** Encapsulates basic calculations functionality that both NCC and SSM use *(commonCalc, commonCalcNCC, commonCalcSSM)*
- ❑ **Net Count Calculation Module** *(ncc)*: Output are net counts for each ROI
- ❑ **Standard Spectra Method Module (ssm):** Output are concentration and uncertainties for each radioxenen

DESCRIPTION

Stores incoming email messages in a temporary directory

Stores email messages in a permanent directory and records the message pointer in the database

Forwards message either to *ParseData* or *MessageAlert*

Messages are parsed into the database or an email is send to specified users in case of failure

RN message content is parsed into the database and written to filesystems. If the Message Subsystem is not the input source, *rms_sainput* is called by the *rms_pipeline* script. *rms_interval_insert* inserts records into the interval table for the workflow GUI.
rms_pl_check is an executable used in *rms_pipeline* and *rms_process* to impose restrictions on which samples will be analyzed or categorized.
If the message contains PHD, then *rms_process* is called to analyze it.
ARSA needs *bg_analyze*;
ARIX needs *gated_analyze* (to be developed);
SAUNA needs *bg_analyze*;
SPALAX needs Xe-Saint (future) and Aatami (now) but is also automatically processed with rms_analyze (results are not exact).

DESCRIPTION OF PROCESSING STEPS WITHIN *rms_analyze*

- ❑ **Retrieve Spectral Data:** Sample, spectrum, energy, efficiency, resolution and processing data are retrieved from a binary file in Canberra format (`.cnf`)
- ❑ **Peak Locate, Peak Area:** A first peak search is performed. Peak locations and peak areas are calculated. Once several peaks are found , the analysis uses these to determine the detector calibration. If no peaks are found analyze fails and exits.
- ❑ **Calibration Update:** Due to changes in the state of a detector, for example gain drifting over time, the initial equations used for the 1st peak search, peak locate may not accurately represent the current detector state. New equation coefficients are derived by attempting to match a list of expected peaks (baselines) to those found in the sample spectrum.
- ❑ **2nd Peak Locate, Peak Area** is performed with the updated coefficients
- ❑ **Lookup Tolerance:** During this step peaks are matched to potential nuclides that may have created them
- ❑ **Efficiency Conversion:** Net counts per second under each peak are converted to a photon emission rate (photons per second)
- ❑ **Nuclide ID + Quantification:** Nuclide libraries (`.nlb` files) contain nuclide data such as half-lives, photon emissions and abundances. If a nuclide is not present in the library it cannot be identified. Depending on the PHD type, either activity values [Bq] for types BLANKPHD, CALIBPHD, DETBKPHD, QCPHD or activity concentrations [Bq/m^3] for type SAMPLEPHD are determined. In a second step weighted values are calculated. When an observed peak has contributions from more than one nuclide an interference correction is performed and weighted activity values and concentrations are calculated via a multiple linear regression. If no interference occurs weighted activities are calculated by a single weighted average of the individual line activities.
- ❑ **MIN Detectable Concentrations:** For every PHD spectrum minimum detectable activities or minimum detectable concentrations for each nuclide in the nuclide library are determined.
- ❑ **Cascade Summing Corrections** have to be done, when multiple gamma rays (or x-rays) are emitted in a decay and are then measured simultaneously by the detector. This can lead to removal or addition of counts from the detected peak area.
- ❑ **Parent/Daughter Calculations:** In some cases, all net counts in a peak of a progeny nuclide are due to the parent's decay into the progeny. If this effect is not corrected, the results show absurdly large atmospheric activity concentrations for the progeny radionuclide. Furthermore a Decay Chain Determination is performed to check if all nuclides in a decay chain are present in the sample and satisfy the half-life-criterion.

Each sample is assigned a category based on whether the analysis results are considered to be normal or abnormal and if the nuclides are natural or man made

During this step various Data Quality and Event Screening tests are performed on the sample to monitor the State of Health of station instruments, the reliability of incoming data and the number and frequency of certain anthropogenic nuclides observed at each station

Automated Radionuclide Reports (ARR) are only processed for FULL SAMPLEPHD data types. Sample information for a given sample ID is collected from the database and arranged in text file. The ARR is available either by request on subscription via the AutoDRM and through the PIDC home page on the Web

Two spectrum qualifiers are used for radionuclide samples: FULL and PREL (preliminary). All FULL samples samples, that have been analyzed run through *rms_mar_auto*. Qualified sample IDs are assigned to users for interactive review as well as permissions on functions to execute during interactive review.

Figure 2: RN processing flow diagram

3.2.9. RN Interactive processing in the Development LAN – overview

There are no regular interactive analysis activities relevant to the processing of the RN data in the Development LAN, but it would be possible to set them up for testing.

3.2.10. RM post analysis activities in the Development LAN - overview

There are no post analysis activities relevant to RN data in the Development LAN.

3.3. RN operations procedures in the Development LAN

3.3.1. Starting the RN processing system

Abstract: instructions on how to start the automatic RN processing system

Additional information:
None.

Procedure:
No special start-up procedure is normally needed, as *MessageGet* feeds each incoming message to ParseData, which writes information to the IDCX.**msgdisc** table and in case of a PHD message launches the *rms_pipeline* script. The processing for the other data types (MET, SOH …) is restricted mainly to import and export.

3.3.2. Shut down of the RN processing system

Abstract: instructions on how to stop the automatic RN processing system

Additional information:
None.

Procedure:
RN processing is not normally shut down as it runs fully automatically, receives input data from the SHI message subsystem and *rms_pipeline* automatically terminates processing of a sample in case of success or failure.

3.3.3. Starting and stopping cron jobs

Abstract: instructions on starting and stopping `cron` jobs.

Additional information:
None.

Procedure:
There are two `crontab` entries - `start_rmsman_migration.crontab`, `start_rmsauto_migration.crontab` - required for the data migration and data purging procedures available in `/dvl/software/rn/Migrate/Data/cron` which can be activated on demand as described in
`/dvl/software/rn/MigrateData/cron/cronjobs.readme`.
See [IDC/DEVLAN/OG-SHI] section 3.3.16 for instructions on how to control data migration and purging.

3.4. RN monitoring procedures

3.4.1. Monitoring the automatic processing workflow

Abstract:
The RN automatic data processing requires regular monitoring in order to assure correctness of products issued. The monitoring practices are described below.

Additional information:
None.

Procedure:
Perform the following routine monitoring tasks on a daily basis to verify the operational status of the Development LAN processing software and to detect any problems as early as possible.

- Open a window and login as `auto` on `eldey`:

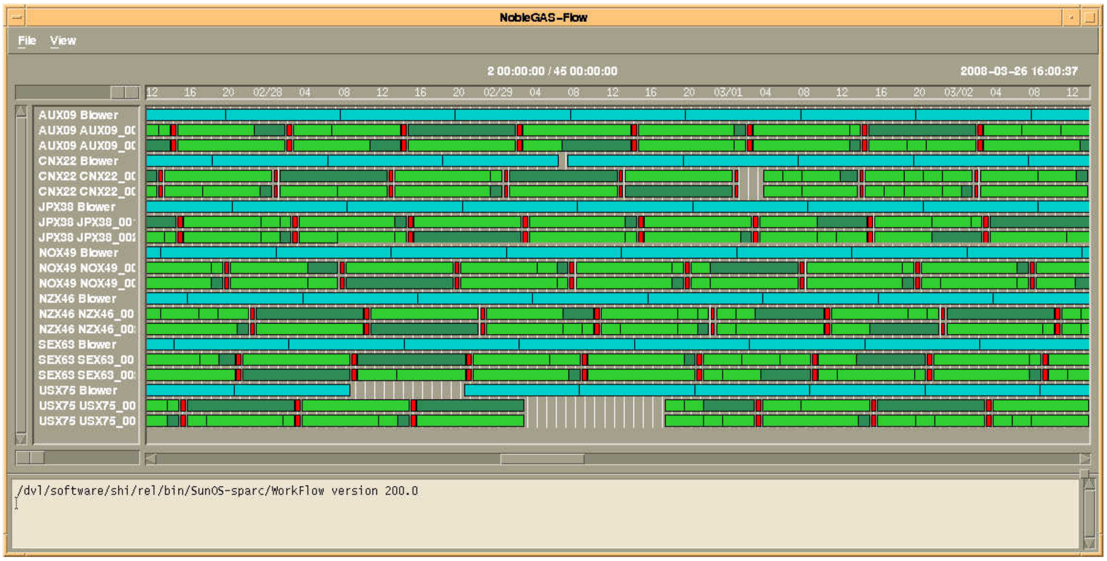
```
ssh auto@eldey
```

- Open two *WorkFlow* windows to check the status for the noble gas and particulate stations:

```
/dvl/software/shi/rel/bin/WorkFlow -name NobleGAS-Flow \
    par=/dvl/software/rn/configs/parmfiles/idcRMS_noble_gas_WF.par
/dvl/software/shi/rel/bin/WorkFlow -name Particulate-Flow \
    par=/dvl/software/rn/configs/parmfiles/idcRMS_particulate_WF.par
```

Currently samples for one particulate station (DEP33) and all noble gas stations (AUX09, CNX22, JPX38, NOX49, NZX46, SEX63, USX75) are forwarded from the Operations LAN to the Development LAN. Additional stations can be installed using the procedure described in section 3.5.2.

(1) After *WorkFlow* screens are displayed it is necessary to check the status of the data acquisition. *WorkFlow* (Figure 3) displays the state of the detector and blower for each station and time interval.



```
/dvl/software/shi/rel/bin/SunOS-sparc/WorkFlow version 200.0
```

Figure 3: Noble gas station WorkFlow snapshot

Blower: The light blue line indicates that the blower (draws air to the sampler) was running and no line means that the blower was off. No action has to be taken on the Dev LAN regarding the blower states.

Detector: Missing intervals or gaps show that no data has been received for a specific station or the automatic processing has failed. Known possible reasons are:

(a) The SHI message subsystem is down:

- Check the SHI message subsystem (see [IDC/DEVLAN/OG-SHI], section 3.4.8).

(b) PHD message did not successfully pass the processing pipeline:

- Check the content of messages directory `/dvl/data/shared/messages/<yyyy>/<jjj>` to verify that RN data messages from the station are received and stored on the Development LAN. Also IDCX**.msgdisc** table content may provide additional information in this regard. If there are messages in status RUNNING and there is an increasing number of messages in status RECEIVED a message-processing queue is blocked. This typically occurs after a crash of an application, which unsuccessfully tried to process a message and left it in status RUNNING. Update the message status from RUNNING to RECEIVED in order to re-process the message, or to FAILED in order to avoid re-processing.
- Check the `invalidmail` directory `/dvl/data/shared/messages/invalidmail` to verify that the sender is successfully authenticated and RN data messages are not rejected.
- Check the log files of the input pipeline processing chain in `/dvl/logs/rn/pipeline` for any processing errors starting with the file `input_<yyyymmdd>.log` where `yyyymmdd` corresponds to the date of interest.
- Check the log files of the automated analysis in `/dvl/logs/rn/analyze` for the particulate samples and in `/dvl/logs/rn/xanalyze` for the noble gas samples to identify any processing errors.

(c) Data are not being forwarded from the Operations LAN to the Development LAN:

- Check the recent procmail logs in /ops/logs/rn/procmail to determine if RN data messages are forwarded to the Development LAN.
- Check the invalidmail directory in /ops/logs/rn/procmail/INVALIDMAIL to determine if procmail rejects any forwarded RN data messages.
- Contact a processing engineer responsible for the IDC Operations LAN to determine if RN data messages are sent to the Development LAN.
- Contact the system administrator to determine if there are any general email problems in the system.

(d) The station does not send data:

- Possible reasons: Station hardware or software failures, station maintenance, network interruptions. Useful information regarding the current status of a station can be derived from the IMS Report System (IRS):
  `http://irs.gci.ctbto.org/`

(2) The next step in monitoring activities is to check sample types received. *WorkFlow* displays the sample types received for each detector and time interval. The sample types are associated with the interval states and colours as follows:

Light green	Preliminary sample (SPHDP) count, 4-hours
Dark green	Full sample (SPHDF) count, 23.5-hours
Salmon	BlankPHD sample count
Magenta	CALIBPHD sample count
Pink	DETBKPHD sample count
Red	QCPHD sample count
Orange	SOH message

(3) Check the current log files in the following directories from time to time, and especially after implementation of new software releases:

```
/dvl/logs/rn/pipeline
/dvl/logs/rn/analyze
/dvl/logs/rn/xanalyze
/dvl/logs/rn/categorize
/dvl/logs/rn/xcategorize
/dvl/logs/rn/DBflags
/dvl/logs/rn/mar
```

3.4.2. Monitoring the processing status in the database

Abstract:
The processing status of the system is to be checked in the database.

Additional information:
[IDC5.1.1] Database Schema,
[IDC5.1.3] Configuration of PIDC Databases.

Procedure:

The primary tool to monitor the data processing status is *WorkFlow*. An alternative monitoring method is to directly check the status of the database tables using *SQL* queries. The interactive monitoring tools offer a quick overview of the system status, while *SQL* queries will reveal more detailed information to trace specific issues across different processing applications or subsystems. Furthermore, most RN database tables, and SHI tables used as interface to the RN software, e.g. IDCX**.msgdisc**, cannot be monitored with the existing interactive tools.
A good understanding of the RN database schema and the entity relationships between the database tables is very helpful to sum up the information from different tables and track the data flow and the processing information through the system. Monitoring the database tables often helps to determine the origin of some specific processing problems, since almost all RN applications insert status information into various database tables.

The most relevant tables for automatic processing are the **msgdisc** table in the IDCX account and **gards_processing_errors**, **gards_data_log and gards_sample_data** in the rmsauto account. They should be checked whenever there is any problem with data reception or automatic processing.

- To check the **msgdisc** table open a window and login as user auto to eldey:

  ```
  ssh auto@eldey
  sqlwrap IDCXDB
  ```

- To check the RN tables, open a window, login on any host in the Development LAN and connect to the rmsauto database account:

  ```
  sqlplus rmsauto/${password}@idcdev
  ```

  (a) The following *SQL* statement can be used to derive all processing errors from **gards_processing_errors** which occurred during the last 24 hours:

  ```
  select * from gards_processing_errors where moddate > sysdate -1;
  ```

  (b) To check if all data types were processed within the last day check the **gards_data_log** table:

  ```
  select * from gards_data_log where moddate > sysdate -1;
  ```

  (c) The table **gards_sample_data** contains the headers of the PHD message files as well as the information on the location of the stored messages. Information derived from this table may be helpful when processing errors occur during the automatic data analysis. The most common failures are related to malformed incoming data messages. The statement below shows all relevant sample information for a given sample_id.

  ```
  select * from gards_sample_data where sample_id = <sample_id>;
  ```

  The sample_id can be directly derived from the name of the analysis log files: <sample_id>.log

3.5. RN maintenance procedures

3.5.1. Interpreting log and error message

Abstract:
Log files are to be checked for the processing status of various RN applications.

Additional information:
None.

Procedure:
The logging area for all RN processing applications is `/dvl/logs/rn`. All automatic and interactive RN applications write their log files into corresponding subdirectories.

3.5.1.1. *Routine checks and monitoring*

Check in regular intervals, that the log file system has sufficient free space for the next daily log files. The log files are not automatically purged, so that the total space used in the log file partition constantly grows. Manually purge old log files to make space available if necessary.

3.5.1.2. *Interpreting error messages and resolving problems*

The specific log and error messages vary for each application. Refer to the RN IDC software documentation for application-specific information. Error messages may also indicate the unavailability of system resources, e.g., if the database connection is unavailable, directories or files cannot be accessed; the network connection is unavailable, etc. Such problems may be transient, i.e., they will disappear after a short time (e.g., if too many processes concurrently tried to obtain the same system resources), or they may be persistent. Manually check the availability of the relevant system resources. If a persistent problem related to system resources is determined contact the system administrator to diagnose the specific problem.

Error messages may indicate the inability of an application to process specific data. In this case the problem can be data-related (e.g., corrupted data file) or software-related. To further determine the nature of the problem try to reproduce it for different data and determine the specific features of the data that may trigger the problem. If the problem is clearly data-related, it should be logged and reported for further assessment. If the problem appears to be related to a software defect, report the software problem. If the problem can be related to specific features, which may routinely occur in regular data intervals, report an enhancement request. See section 2.6.5 of [IDC/DECLAN/OG-SHI] for instruction on reporting problems.

In addition to the application log files inspection of the system *syslog* files in the mass store together with a system administrator may be helpful in specific cases.

3.5.2. Installation of new stations

3.5.2.1. *Manual noble gas station installation*

Abstract:
A new noble gas station is to be installed in the system.

Additional information:
[CIN-01462] Installation of new IMS Radionuclide stations

Procedure:
The installation of a new RN station consists of a combination of database and configuration changes that depend on the type of a station to be installed. The generic station installation CIN-01462 lists the changes to be made.

The station configuration data for stations to be installed or upgraded is maintained by the IMS division and is provided to the IDC. New RN stations are first installed in the Testbed and later in the Operations LAN. The Development LAN station configuration generally follows the Testbed configuration, and new stations are typically installed on the Development LAN after they have been installed on the Testbed. The development LAN operator has to contact the Processing Engineers to obtain the configuration data for the station to be installed.

The specific procedure below applies to the installation of a RN station which delivers noble gas beta-gamma coincidence data from ARSA and SAUNA systems. Station information and initial processing parameters for the new noble gas station have to be inserted into the following radionuclide database tables:

RMSMAN.**gards_stations**
RMSMAN.**gards_detectors**
RMSMAN.**gards_stadet**
RMSMAN.**gards_bg_config_params**
IDCX**.datauser**

- Connect to the RMSMAN database account on the Development LAN:

  ```
  sqlplus rmsman@idcdev
  ```

- Entries in **gards_stations**:

  Get the last `station_id` from **gards_stations** and use the next available ID for the new station:

  ```
  select max(station_id) from gards_stations
  ```

  Insert the station information for the new station. Example values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into
  gards_stations(STATION_ID,STATION_CODE,COUNTRY_CODE,
  TYPE,DESCRIPTION,LAT,LON,ELEVATION,DATE_BEGIN,MODDATE) VALUES
  (<585>,'<OSX91>','<A>','<SAUNA>','<Noble Gas Experi. OSI
  Seibersdorf>',<47.9750>,<16.5080>,<186>,sysdate,sysdate);
  ```

- Entries in **gards_detectors**:

  Get the last `detector_id` from **gards_detectors** and use the next available ID for the new detector:

  ```
  select max(detector_id) from gards_detectors;
  ```

  Insert the detector information for each detector. Example values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_detectors
  (DETECTOR_ID,DETECTOR_CODE,DESCRIPTION,
  TYPE,DATE_BEGIN,STATUS,MODDATE) VALUES
  ```

```
(<550>,'<OSX91_001>','<Detector #1 Seibersdorf, Austria>',
sysdate,'Y',sysdate);
```

- Entries in **gards_stadet**:

  Insert `detector_id`/`station_id` combinations for each detector using the same IDs as for the previous insert queries. Example values in *italic* font and between "<>" have to be replaced by actual values:

  ```
  insert into gards_stadet (STATION_ID,DETECTOR_ID,MODDATE)
  VALUES (<585>,<550>,sysdate);
  ```

- Entries in **gards_bg_config_params**:

  Insert configuration parameters for each detector. Example values in *italic* font and between "<>" have to be replaced by actual values:

  ```
  insert into gards_bg_config_params VALUES
  (<550>,1.6449,0,.4,10,43200,43200,172800,1,.4,1,1,1,40,
  101001,.087,43200,3,10,1,1,100000,sysdate);
  ```

- Entries in IDCX.**datauser**:

  Connect to the `IDCXDB` on the Development LAN:

  ```
  ssh cmss@eldey
  sqlwrap IDCXDB
  ```

  Get the last `userid` from IDCX**.datauser** and use the next available ID for the new station address:

  ```
  select max(userid) from datauser
  ```

  Insert the sender e-mail address of the new station. Example values in *italics* and between "<>" have to be replaced by actual values:

  ```
  insert into datauser (USERID,USERNAME,DOMAIN,MSGTYPE,STATUS,
  PRIORITY,EMAILLIMIT,SERVICETIME,LDDATE) VALUES
  (<5103>,'<ws03>','<ctbto.org>','DATA','ACTIVE',
  1,1048576,1.000E+10,sysdate);
  ```

  An alternative way of populating the IDCX**.datauser** table with the entries relevant to a new station installation is to run the *datauser* script located in the `scripts/bin` directory. In order to see the detailed instruction on how to use it type in command prompt: `datauser -h[elp]`

- Add the station code as user rmscm to the WorkFlow parameter file:

  ```
  ssh rmscm@eldey
  /dvl/software/rn/configs/parmfiles/idcRMS_noble_gas_WF.par
  ```

3.5.2.2. *Manual installation of a particulate RN station*

The specific procedure below applies to the installation of a particulate RN station on Development LAN. Station information and initial processing parameters for the new particulate station have to be inserted into the following radionuclide database tables:

RMSMAN.**gards_stations**
RMSMAN.**gards_detectors**
RMSMAN.**gards_stadet**
RMSMAN.**gards_proc_params_template**
RMSMAN.**gards_update_params_template**
RMSMAN.**gards_update_reflines**

IDCX**.datauser**

- Connect to the RMSMAN database account on the Development LAN:

  ```
  sqlplus rmsman@idcdev
  ```

- Entries in **gards_stations**:

  Get the last `station_id` from **gards_stations** and use the next available ID for the new station:

  ```
  select max(station_id) from gards_stations
  ```

  Insert the station information for the new station. Example values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into
  gards_stations(STATION_ID,STATION_CODE,COUNTRY_CODE,
  DESCRIPTION,LAT,LON,ELEVATION,DATE_BEGIN,MODDATE) VALUES
  (<518>,'<FRP27>','<FR>','<Papeete, Tahiti>’,< -17.5760>,
  < -149.5670>,<300>,sysdate,sysdate);
  ```

- Entries in **gards_detectors**:

  Get the last `detector_id` from **gards_detectors** and use the next available ID for the new detector:

  ```
  select max(detector_id) from gards_detectors;
  ```

  Insert the detector information for each detector. Example for values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_detectors
  (DETECTOR_ID,DETECTOR_CODE,DESCRIPTION,LAT,LON,
  TYPE,CHANNELS,RATED_EFFICIENCY,RATED_RESOLUTION,ECAL_RANGE_MAX,
  DATE_BEGIN,STATUS,MODDATE) VALUES
  (<437>,'<FRP27_001>','<Detector in Papetee, Tahiti>’,<CoaxN>,
  <8192>,<50>,<2.1>,<2700>, sysdate,'Y',sysdate);
  ```

- Entries in **gards_stadet**:

  Insert `detector_id`/`station_id` combinations for each detector using the same IDs as for the previous insert queries. Example values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_stadet (STATION_ID,DETECTOR_ID,MODDATE)
  VALUES (<518>,<437>,sysdate);
  ```

- Entries in **gards_proc_params_template**:

  Insert processing parameters for each data_type (B,C,D,Q,S) and each spectral qualifier (PREL,FULL). Example for adding one record is listed below, values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_proc_params_template (STATION_ID,DETECTOR_ID,
  SAMPLE_TYPE,DATA_TYPE,SPECTRAL_QUALIFIER,BEGIN_DATE,DO_BACK,BAC
  K_DATA_TYPE,NUCLIDE_LIB,MDA_LEVEL,NID_CONFID,SQUANT_ERR,BUILDTY
  PE,PEAK_SENSE,PEAK_START,PEAK_END,FWHM_MULT_WIDTH,LEFT_FWHM_LIM
  ,RIGHT_FWHM_LIM,BACK_CHAN,BACK_TYPE,FIT_SINGLETS,CRIT_LEVEL,FIX
  _FWHM,AREA_REJECT,DO_CSC,MODDATE)  VALUES  (<518>,<437>,'<P>',
  '<S>','<PREL>',sysdate,<0>,'<B>',
  '/home/misc/rmsops/target/genie/idc_lib_p_v2_1.nlb',
  <5>,<.25>,<2>,'<BUILD>',<3>,<42>,<3000>,<4>,<2>,<2>,<3>,'STEP',
  <1>,<1>,<0>,<1>,<0>,sysdate);
  ```

- Entries in **gards_update_params_template**:

  Insert update parameters for each detector_id. Example values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_update_params_template (DETECTOR_ID,
  GAINSHIFT,ZEROSHIFT,AREA_LIM,USE_WEIGHT,USE_MULT,F_LINEAR,BOOTS
  TRAP,MIN_LOOKUP,RER_INTERCEPT,RER_SLOPE,DO_RERU,MODDATE) VALUES
  (<437>,<.1>,<.5>,<100>,<1>,<1>,<0>,<0>,<0.2>,<1.5>,<0.03>,<1>,
  sysdate);
  ```

- Entries in **gards_update_reflines**:

  Insert reference peak energy for each data_type (B,C,D,Q,S) and each spectral qualifier (PREL,FULL) → ~ more than 100 records. Example for adding one record is listed below, values in *italic* font and between “<>” have to be replaced by actual values:

  ```
  insert into gards_update_reflines (STATION_ID,DETECTOR_ID,
  REFPEAK_ENERGY,DATAY_TYPE,SPECTRAL_QUALIFIER, MODDATE) VALUES
  (<518>,<437>,<1093.9,’<S>’,’<PREL>’,sysdate);
  ```

- Entries in IDCX.**datauser**:

  Connect to the `IDCXDB` on the Development LAN:

  ```
  ssh cmss@eldey
  sqlwrap IDCXDB
  ```

  Get the last `userid` from IDCX.**datauser** and use the next available ID for the new station address:

  ```
  select max(userid) from datauser
  ```

  Insert the sender e-mail address of the new station. Example values in *italics* and between “<>” have to be replaced by actual values:

  ```
  insert into datauser (USERID,USERNAME,DOMAIN,MSGTYPE,STATUS,
  PRIORITY,EMAILLIMIT,SERVICETIME,LDDATE) VALUES
  (<5103>,'<ws03>','<ctbto.org>','DATA','ACTIVE',
  1,1048576,1.000E+10,sysdate);
  ```

  An alternative way of populating the IDCX.**datauser** table with the entries relevant to a new station installation is to run the *datauser* script located in the `scripts/bin` directory. In order to see the detailed instruction on how to use it type in command prompt: `datauser -h[elp]`

- Add the station code as user rmscm to the WorkFlow parameter file:

```
ssh rmscm@eldey
/dvl/software/rn/configs/parmfiles/idcRMS_particulate_WF.par
```

The following instructions are used in the Operations and Testbed LAN for the purpose of installation of particulate RN stations.

Use *rms_dwb* as `rmsman` database user:

- `source ~rmsops/.cshrc`
- start *rms_dwb*
- insert new/update old Point-Of-Contact information
- *Save* and *Exit*
- click on *Stations* button
- check country information
- insert the station information and note the assigned station number
- *Save* and *Exit*
- click on *Detectors* button
- insert a new detector and note the assigned detector number
- make sure that detector status is set to 'Y'
- *Save* and *Exit*
- exit *rms_dwb*
- connect as `rmsman` user to the database
- run the `~laban/insert_new_detector_info.sql` script
- type in station and detector IDs when prompted to do so
- check **gards_stadet** table and update it with the new information relevant to the station and the detector
- commit changes and quit.

3.6. RN troubleshooting procedures

3.6.1. bg_analyze troubleshooting

Abstract:
This section describes operator actions to respond to various undesirable observations in connection with execution of *bg_analyze*

Additional information:
None.

Observation: *bg_analyze* shows unexpected processing result

Procedure:
When *bg_analyze* shows unexpected processing results for data messages, which have been re-sent it is necessary to check the log file in `/dvl/logs/rn/xanalyze` directory.

If the message was re-sent the transmit date and/or transmit time (record 5 in the header block) has to be set to the current date in order to enable re-processing of the data message. Parameters which are related to the measurement identification, sample reference identification, spectrum qualifier or sample geometry should not ever be changed. Otherwise this would lead to incorrect analysis results.

It is possible that references to detector background or other samples do not exist any more in the system, specifically for old messages which may be re-sent. In such a case the development LAN operator has to investigate the sample-specific log-file in `/dvl/logs/rn/xanalyze` to restore all samples which are linked to the individual message from the archive or from a backup copy.

Observation: *bg_analyze* fails when sample order is not correct

Procedure:
If *bg_analyze* fails due to an incorrect sample order it is necessary to check the log file directory `/dvl/logs/rn/xanalyze` entries. *bg_analyze* requires samples to be sent in the following order:

1. QC
2. DETBK
3. GASBK
4. SAMPLES

If there is no QC sample present or the sample itself has an acquisition date prior to the date of the QC sample, *bg_analyze* will terminate with an error message saying "`fatal error in function readHistogramSettings - unexpected number of rows returned`". This error message is not very specific and can be generated due to various reasons, for example when the data blocks in the input file are not correctly formatted, data blocks contain a wrong number of rows or corrupt data was written to RN database tables during processing.

If no DETBK sample is present the default detector background will be used and no background subtraction will be performed. Furthermore the spectrum is set to background and concentrations are not calculated. This is a known bug in the current version 1.0.5 of *bg_analyze*.

If there is no GASBK sample present for a currently processed sample, *bg_analyze* will terminate with the error "`fatal error in readGasBkngdSampleID- unexpected number of rows returned - No gas background found`".

Observation: *bg_analyze* fails due to a fatal error in function *readTimes*

Procedure:
Log file location: `/dvl/logs/rn/xanalyze`
bg_analyze terminates with the error message "`fatal error in function readTimes`". This error occurs when the measurement time information has been manually manipulated and the acquisition stop time has a date prior to acquisition start time.

Observation: *bg_analyze* fails due to a fatal error in function *checkSampleID*

Procedure:
Log file location: `/dvl/logs/rn/xanalyze`

bg_analyze terminates with the error message "`fatal error in function checkSampleID - Data type unknown`".
Check if the sample is of type `CALIBPHD`. *bg_analyze* is not able to process samples of this type.

Observation*:* *bg_analyze* fails due to a fatal error in function *loadSpectras*

Procedure:
Log file location: `/dvl/logs/rn/xanalyze`
bg_analyze terminates with the error message "`fatal error in function loadSpectras - Cannot open specified file`".
One or more histogram files could not be read. Check the corresponding histogram file in `/home/misc/rmsops/data/histogram/<date>`.

Observation: *bg_analyze* fails due to a fatal error in function *loadDataNCC*

Procedure:
Log file location: `/dvl/logs/rn/xanalyze`
bg_analyze terminates with the error message "`fatal error in function loadDataNCC - Failed loading NCC data from database`".

This error occurs when the `G_ENERGY_STOP` value is less than the `G_ENERGY_START` value in the **gards_roi_limits** table. The ROI limits in the database can be checked with the following *SQL* statement:

```
sqlplus rmsauto@idcdev
select sample_id,g_energy_start,g_energy_stop,moddate from
gards_roi_limits where ROI=3 and moddate > sysdate - <enter
appropriate time span> order by moddate;
```

This error can occur when a faulty *rms_input* version is promoted to the runtime system. In one known case changes in the function `librms_input` caused this fatal error.

3.6.2. autoSaint troubleshooting

See [IDC/autoSaint/SUG] Auto-saint Software User Guide, section 7.

More details will be added here when experience with autoSaint has been gathered on the Development LAN.

3.6.3. rms_input troubleshooting

The following problems are the most common ones:

- MSG format not compliant to "Formats and Protocols for Messages"

```
Error (-1) inserting soh header info for soh_id 38045 --> 20084-
20060130140920 ***** UNRECOGNIZED SECTION HEADER FOUND IN SOH
MESSAGE 20084-20060130140920 VERIFY THE MESSAGE. *****  20084-
20060130140920 Section: PowerSupply2
```

- Attempt to parse in the same sample twice

```
14074-20060215131219 Error with SQL statement: 14074-
20060215131219 ORA-00001: unique constraint
(RMSAUTO.UQ_SAMPLE_DATA) violated 14074-20060215131219 insert
into gards_sample_data
```

- Wrong or missing station or detector configuration

```
14034-20060215135507 ERROR: error getting station or detector
info for station name: ARX01
```

or

```
14112-20060215135511 ERROR retrieving STATION ID for   14112-
20060215135511 Station Code: ARX01
```

- Full spectrum file system problem

```
16756-20060428000852 ERROR: Error setting up date directory for
spectrum files: path = /home/misc/rmsops/data/spectrum  16756-
20060428000852 ERROR: Error parsing g spectrum
```

3.6.4. Reprocessing of the particulate station data with customized parameters

In order to re-process incoming data for particulate station with customized energy, resolution or energy tolerance parameters it necessary to run as user `auto` the following command

```
rms_analyze -c $e_coef1 $e_coef2 $e_coef3 $e_coef4 -r $r_coef1\
$r_coef2 -t $et_coef1 -s $Sample_ID
```

3.6.5. Revoking released RN samples

Sometimes RN analysts issue RRR products by mistake due to various reasons. In order to prevent the distribution of wrongly issued products a responsible processing engineer or a Development LAN operator has 30 minutes to intervene and stop the process. To do this it is necessary to connect to the `rmsman` database account and run the `revoke_release_sample.sql` script which is located in `~laban` directory.

After that the relevant row in the **gard_sample_status** table needs to be checked and updated if needed. The last step of the process is to remove the file from a temporary directory in `~rmsops/products/temp_rrr/YY_Mon_DD/rrr_$Sample_ID.1`. The exact file name and location can be identified from the **gards_product** table.

3.6.6. Changing Calibration Reflines for a detector

Use *rms_dwb* as `rmsman` database user.

- Click Reflines->station->detector->Cancel
- Click on relevant energy and modify it as requested by an analyst
- Save change
- Edit 'masterlist'
- Insert master line for each data type "insert multiple reflines"

3.6.7. Analysis and logging of unknown problems

Abstract:
An unknown problem has occurred in the system.

Additional information:
[IDC3.4.1] Formats and Protocols for Messages
[IDC5.1.1] Database Schema
[IDC5.2.2] IDC Processing of Radionuclide Data Rev. 3
[IDC6.5.10] Radionuclide Software User Manual Rev. 4
[IDC6.5.19] Message Subsystem Software User Manual
[IDC6.5.x] IDC Software User Manuals
[IDC7.x.y] IDC Software Design Documents

Procedure:

Analyze the log files to determine the type of the problem. Reprocess the RN message either by resending it to the Development LAN or by updating the status of the message to `RECEIVED` in the IDCX.**msgdisc** table if there is a chance that the problem was of a transient nature.

Use *rms_logbook* to inform other colleagues about the problem.

Refer to the referenced documentation volumes to determine the category of the problem, then document and report the problem following [IDC/DEVLAN/OG-SHI] section 3.6.5.

APPENDIX A. DESCRIPTION OF RADIONUCLIDE DATABASE TABLES

Due to software changes and enhancements since 2001, the RN database schema as published in the IDC documentation ([IDC-5.1.1Rev3] Part 3) is partially obsolete. As a partial remedy, a current description of all RN tables is published here.

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
ALERT_SEQ	SEQUENCE				
			COPY_STATIONS	TABLE	Sequence that creates autonumbers for ALERT IDs
			CREATE$JAVA$LOB$TABLE	TABLE	
DEBUG	TABLE		DEBUG	TABLE	
FILEPRODUCT	SYNONYM	IDCX	FILEPRODUCT	SYNONYM	
FPDESCRIPTION	SYNONYM	IDCX	FPDESCRIPTION	SYNONYM	
GARDS_ALERTS	TABLE	RMSAUTO	GRADS_ALERTS	SYNONYM	Contains data describing all radionuclide alert messages received
GARDS_AUTO_SAMPLE_CAT	TABLE				Contains original categorization values from automatic processing
GARDS_AUX_LIB	SYNONYM	RMSMAN	GARDS_AUX_LIB	TABLE	Provides additional radionuclide information used in the parent-progeny calculation
GARDS_AUX_LINES_LIB	SYNONYM	RMSMAN	GARDS_AUX_LINES_LIB	TABLE	Provides additional radionuclide line information used in the parent-progeny calculation
GARDS_B_ENERGY_PAIRS	TABLE		GARDS_B_ENERGY_PAIRS	TABLE	Contains the energy calibration pairs information associated with the beta axis of the spectrum used in rms_xanalyze. The values in the GARDS_B_ENERGY_PAIRS and the GARDS_B_ENERGY_PAIRS_ORIG tables are identical
GARDS_B_ENERGY_PAIRS_ORIG	TABLE		GARDS_B_ENERGY_PAIRS_ORIG	TABLE	Contains the original energy calibration pairs information associated with the beta axis of the spectrum as specified in the PHD file. The values in the GARDS_B_ENERGY_PAIRS and the GARDS_B_ENERGY_PAIRS_ORIG tables are identical. This table is not used by the *rms_xanalyze* process
GARDS_B_RESOLUTION_PAIRS	TABLE		GARDS_B_RESOLUTION_PAIRS	TABLE	Contains the resolution calibration pairs information associated with the beta axis of the spectrum. This table is not used by the *rms_xanalyze* process. The values in the GARDS_B_RESOLUTION_PAIRS and the GARDS_B_RESOLUTION_PAIRS_ORIG tables are identical

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	**TYPE**	**SYNONYM FROM**	**TABLE_NAME**	**TYPE**	
GARDS_B_RESOLUTION_PAIRS_ORIG	**TABLE**		**GARDS_B_RESOLUTION_PAIRS_ORIG**	**TABLE**	Contains the original resolution calibration pairs information associated with the beta axis of the spectrum as specified in the PHD file. This table is not used by the *rms_xanalyze* process. The values in the GARDS_B_RESOLUTION_PAIRS and the GARDS_B_RESOLUTION_PAIRS_ORIG tables are identical
GARDS_BASELINE	**SYNONYM**	**RMSMAN**	**GARDS_BASELINE**	**TABLE**	The table contains the receipt for the baseline calculation as general or function of a detector_id and type of sample
GARDS_BG_CONFIG_PARAMS	**SYNONYM**	**RMSMAN**	**GARDS_BG_CONFIG_PARAMS**	**TABLE**	
GARDS_BG_DETECTOR_STD_SPECTRA	**TABLE**	**RMSAUTO**	**GARDS_BG_DETECTOR_STD_SPECTRA**	**SYNONYM**	Contains configuration parameters for each detector as input for *bg_analyze*
GARDS_BG_EFFICIENCY_PAIRS	**TABLE**		**GARDS_BG_EFFICIENCY_PAIRS**	**TABLE**	Contains the detection efficiency associated with a beta-gamma coincidence event as specified in the PHD file
GARDS_BG_ENERGY_CAL	**TABLE**		**GARDS_BG_ENERGY_CAL**	**TABLE**	Contains energy-to-channel ratios of a sample
GARDS_BG_ISOTOPE_CONCS	**TABLE**		**GARDS_BG_ISOTOPE_CONCS**	**TABLE**	Contains isotope concentrations for each nuclide in each sample
GARDS_BG_PROC_PARAMS	**TABLE**		**GARDS_BG_PROC_PARAMS**	**TABLE**	Contains specific processing parameters for each sample
GARDS_BG_PROC_PARAMS_ROI	**TABLE**		**GARDS_BG_PROC_PARAMS_ROI**	**TABLE**	Contains specific processing parameters for each ROI
GARDS_BG_QC_RESULT	**TABLE**		**GARDS_BG_QC_RESULT**	**TABLE**	Contains quality control results
GARDS_BG_ROI_CONCS	**TABLE**		**GARDS_BG_ROI_CONCS**	**TABLE**	Contains concentrations for each ROI
GARDS_BG_ROI_COUNTS	**TABLE**		**GARDS_BG_ROI_COUNTS**	**TABLE**	Contains counts for each ROI
GARDS_BG_STD_SPECTRA	**TABLE**	**RMSAUTO**	**GARDS_BG_STD_SPECTRA**	**SYNONYM**	Holds all standard spectra. Standard spectra are related to a nuclide through the nucle_id field
GARDS_BG_STD_SPECTRA_RESULT	**TABLE**		**GARDS_BG_STD_SPECTRA_RESULT**	**TABLE**	Holds results calculated for each standard spectra
GARDS_CAT_CRITERIA_TESTS	**SYNONYM**	**RMSMAN**	**GARDS_CAT_CRITERIA_TESTS**	**TABLE**	Contains a list of all category criteria tests. The column active_flag is set to 0 if the test should not be run. A trigger updates the moddate column when any modification is made to another column
GARDS_CAT_TEMPLATE	**SYNONYM**	**RMSMAN**	**GARDS_CAT_TEMPLATE**	**TABLE**	Contains categorization initialization information for typical nuclides

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_CODES	SYNONYM	RMSMAN	GARDS_CODES	TABLE	Contains codes for items used throughout radionuclide monitoring system
GARDS_COMMENTS	TABLE		GARDS_COMMENTS	TABLE	Contains automated analysis and interactive review comments related to peaks in the GARDS_PEAKS table and nuclides in THE GARDS_NUCL_IDED table
GARDS_COMMENTS_DEFS	SYNONYM	RMSMAN	GARDS_COMMENTS_DEFS	TABLE	Contains predefined comments for automated analysis and interactive review
			GARDS_COMMENTS_DEFS_BG	TABLE	
GARDS_COMMENTS_SEQ	SYNONYM	RMSMAN	GARDS_COMMENTS_SEQ	SEQUENCE	Sequence that creates autonumbers for comments
GARDS_CSC_MODCOEFF_LIB	SYNONYM	RMSMAN	GARDS_CSC_MODCOEFF_LIB	TABLE	Contains the cascade summing rations used by *rms_analyze*
GARDS_DATA_LOG	TABLE	RMSAUTO	GARDS_DATA_LOG	SYNONYM	Contains data describing all radionuclide data messages received
GARDS_DBROLE_OWNER	SYNONYM	RMSMAN	GARDS_DBROLE_OWNER	TABLE	Contains information used by the MAR tool to determine the database user that owns the RMS roles
GARDS_DETECTOR_ID	SYNONYM	RMSMAN	GARDS_DETECTOR_ID	SEQUENCE	Sequence that creates autonumbers for DETECTOR IDs
GARDS_DETECTORS	SYNONYM	RMSMAN	GARDS_DETECTORS	TABLE	Contains detector overviews and characteristics
GARDS_DIST_SAMPLE_QUEUE	SYNONYM	RMSMAN	GARDS_DIST_SAMPLE_QUEUE	TABLE	Contains a list of samples and the user role to which they are assigned
GARDS_DL_ID_SEQ	SYNONYM	RMSMAN	GARDS_DL_ID_SEQ	SEQUENCE	Sequence that creates autonumbers for DATA LOG IDs
GARDS_EFFICIENCY_CAL	TABLE		GARDS_EFFICIENCY_CAL	TABLE	Contains the efficiency calibration equation that is computed for each spectra
GARDS_EFFICIENCY_PAIRS	TABLE		GARDS_EFFICIENCY_PAIRS	TABLE	Contains efficiency calibration pairs as specified in the PHD file
GARDS_EFFICIENCY_VSGL_PAIRS	SYNONYM	RMSMAN	GARDS_EFFICIENCY_VSGL_PAIRS	TABLE	The table contains the efficiency pairs simulated from VGSL for a given detector_id
GARDS_ENERGY_CAL	TABLE		GARDS_ENERGY_CAL	TABLE	Contains efficiency calibration coefficients associated with spectra. The equation is calculated during energy calibration update
GARDS_ENERGY_CAL_COV	TABLE		GARDS_ENERGY_CAL_COV	TABLE	The GARDS_ENERGY_CAL_COV table contains the covariance table for each coefficient of the energy calibration from a given sample_id

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_ENERGY_CAL_ORIG	**TABLE**		**GARDS_ENERGY_CAL_ORIG**	**TABLE**	Contains original energy calibration equation information associated with samples (calculated using energy pair data)
GARDS_ENERGY_PAIRS	**TABLE**		**GARDS_ENERGY_PAIRS**	**TABLE**	Contains energy calibration pairs information associated with the gamma axis of the spectrum
GARDS_ENERGY_PAIRS_ORIG	**TABLE**		**GARDS_ENERGY_PAIRS_ORIG**	**TABLE**	Contains energy calibration pairs information associated with the gamma axis of the spectrum. These values are as specified in the PHD file
GARDS_ENVIRONMENT	**TABLE**	**RMSMAN**	**GARDS_ENVIRONMENT**	**SYNONYM**	Contains atmospheric conditions and related sample information. This table is applicable to PHDs sent in IMS1.0 format or earlier. It does not apply to messages in IMS2.0 format or later
GARDS_FLAGS	**TABLE**		**GARDS_FLAGS**	**TABLE**	Contains the name and thresholds for each of the tests run during *rms_DBflags*
GARDS_FPE	**TABLE**		**GARDS_FPE**	**TABLE**	Contains one record for each sample associated with a fission product event
GARDS_FPID_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_FPID_SEQ**	**SEQUENCE**	Sequence that creates autonumbers for fission product IDs
GARDS_GRANT_ID_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_GRANT_ID_SEQ**	**SEQUENCE**	Sequence that creates autonumbers for ???
GARDS_HISTOGRAM	**TABLE**	**RMSAUTO**	**GARDS_HISTOGRAM**	**SYNONYM**	Contains information regarding the 3-D beta-gamma coincidence sample data
GARDS_INTERVAL	**TABLE**	**RMSAUTO**	**GARDS_INTERVAL**	**SYNONYM**	Contains data used to update the RMS data workflow
GARDS_IRF	**SYNONYM**	**RMSMAN**	**GARDS_IRF**	**TABLE**	The table contains the isotope response function, its error and the summing corrections as a function of the energy for a given Nuclide and a given Detector
GARDS_LAB_CATEGORY_DESCRIPTION	**SYNONYM**	**RMSMAN**	**GARDS_LAB_CATEGORY_DESCRIPTION**	**TABLE**	Holds laboratory specific descriptions
GARDS_MDAS2REPORT	**SYNONYM**	**RMSMAN**	**GARDS_MDAS2REPORT**	**TABLE**	Contains the list of nuclides which have their minimum detectable activities reported in radionuclide reports
GARDS_MET_DATA	**TABLE**	**RMSAUTO**	**GARDS_MET_DATA**	**SYNONYM**	Contains station local meteorological data
GARDS_METHOD_ID_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_METHOD_ID_SEQ**	**SEQUENCE**	
GARDS_NIC	**TABLE**		**GARDS_NIC**	**TABLE**	Empty
GARDS_NOTIFY	**TABLE**		**GARDS_NOTIFY**	**TABLE**	Contains contact information for specific events. When the event occurs a message is automatically

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
					sent to the *email_addr* of the specified recipient
GARDS_NUCL_IDED	**TABLE**		**GARDS_NUCL_IDED**	**TABLE**	Contains information regarding nuclides identified during analysis
GARDS_NUCL_IDED_ORIG	**TABLE**		**GARDS_NUCL_IDED_ORIG**	**TABLE**	Contains information regarding nuclides identified during automated analysis
GARDS_NUCL_LIB	**SYNONYM**	**RMSMAN**	**GARDS_NUCL_LIB**	**TABLE**	Contains nuclide library information used in particulate nuclide identification process
GARDS_NUCL_LINES_IDED	**TABLE**		**GARDS_NUCL_LINES_IDED**	**TABLE**	Contains auxiliary information regarding lines identified
GARDS_NUCL_LINES_IDED_ORIG	**TABLE**		**GARDS_NUCL_LINES_IDED_ORIG**	**TABLE**	Contains information regarding lines identified during automated analysis
GARDS_NUCL_LINES_LIB	**SYNONYM**	**RMSMAN**	**GARDS_NUCL_LINES_LIB**	**TABLE**	Contains library information used in particulate nuclide identification process
GARDS_NUCL2QUANTIFY	**SYNONYM**	**RMSMAN**	**GARDS_NUCL2QUANTIFY**	**TABLE**	
GARDS_PEAKS	**TABLE**		**GARDS_PEAKS**	**TABLE**	Contains information regarding peaks identified
GARDS_PEAKS_ORIG	**TABLE**		**GARDS_PEAKS_ORIG**	**TABLE**	Contains information regarding peaks identified during automated analysis
GARDS_PERM_ID_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_PERM_ID_SEQ**	**SEQUENCE**	Sequence that creates autonumbers for PERMISSION IDs
GARDS_PERMISSIONS	**SYNONYM**	**RMSMAN**	**GARDS_PERMISSIONS**	**TABLE**	Contains a list of permissions that can be manipulated by the MAR tool
GARDS_POC	**SYNONYM**	**RMSMAN**	**GARDS_POC**	**TABLE**	Contains information for radionuclide monitoring system point of contact (POC)
GARDS_POCID_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_POCID_SEQ**	**SEQUENCE**	Sequence that creates autonumbers for POINT OF CONTACT IDs
GARDS_PROC_PARAMS_TEMPLATE	**SYNONYM**	**RMSMAN**	**GARDS_PROC_PARAMS_TEMPLATE**	**TABLE**	Template data record that contains parameters used by automatic analysis software for peak search and nuclide identification. These parameters are used unless overridden at the command line
GARDS_PROCESSING_ERRORS	**SYNONYM**	**RMSMAN**	**GARDS_PROCESSING_ERRORS**	**TABLE**	This table contains sample information, if applicable, and message information for RMS messages that have failed processing
GARDS_PRODUCT	**SYNONYM**	**RMSMAN**	**GARDS_PRODUCT**	**TABLE**	Holds temporary reports (last entry Feb 2006)
GARDS_PRODUCT_TYPE	**SYNONYM**	**RMSMAN**	**GARDS_PRODUCT_TYPE**	**TABLE**	Contains report types (probably no longer used...)
GARDS_QAT_CONFIG	**SYNONYM**	**RMSMAN**	**GARDS_QAT_CONFIG**	**TABLE**	Contains *q_time* and *t_time* used by *rms_QAT_auto*

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_QAT_NOTIFY	**SYNONYM**	**RMSMAN**	**GARDS_QAT_NOTIFY**	**TABLE**	Contains comments created by *rms_QAT* (empty)
GARDS_QAT_QUERY_FILTER	**SYNONYM**	**RMSMAN**	**GARDS_QAT_QUERY_FILTER**	**TABLE**	Contains each user's default query in *rms_QAT*
GARDS_QC_RESULTS	**TABLE**		**GARDS_QC_RESULTS**	**TABLE**	The table contains the result of QC performed for the given sample_id
GARDS_QCHISTORY	**TABLE**		**GARDS_QCHISTORY**	**TABLE**	Contains output messages from quality control tests for each sample
GARDS_QCPARAMS	**SYNONYM**	**RMSMAN**	**GARDS_QCPARAMS**	**TABLE**	Contains parameters for quality control processes
GARDS_QCTARGETS	**SYNONYM**	**RMSMAN**	**GARDS_QCTARGETS**	**TABLE**	Contains detector-specific target values for the quality control width and area tests
GARDS_QUERY_RESULTS	**TABLE**		**GARDS_QUERY_RESULTS**	**TABLE**	Contains output of a user initiated query from the event screening tool. This table is purged at the end of the request
GARDS_QVIEWER_SEQ	**SYNONYM**	**RMSMAN**	**GARDS_QVIEWER_SEQ**	**SEQUENCE**	Sequence that creates autonumbers for TRENDVUE SESSION IDs
GARDS_READ_SAMPLE_CAT	**SYNONYM**	**GARDS_SAMPLE_CAT**	**GARDS_READ_SAMPLE_CAT**	**SYNONYM**	Synonym for GARDS_SAMPLE_CAT, which allows *rms_categorize* to read GARDS_SAMPLE_STATUS regardless of how it was started
GARDS_READ_SAMPLE_STATUS	**SYNONYM**	**GARDS_SAMPLE_STATUS**	**GARDS_READ_SAMPLE_STATUS**	**SYNONYM**	This table is a synonym for GARDS_SAMPLE_STATUS, which allows *rms_categorize* to read GARDS_SAMPLE_STATUS regardless of how it was started
GARDS_RECEIPT_LOG	**TABLE**	**RMSAUTO**	**GARDS_RECEIPT_LOG**	**SYNONYM**	This table is used by the *gards_receipt*s utility
GARDS_REFLINE_MASTER	**SYNONYM**	**RMSMAN**	**GARDS_REFLINE_MASTER**	**TABLE**	Contains resolution calibration pairs information associated with spectral pulse height data (PHD)
GARDS_RELEVANT_NUCLIDES	**SYNONYM**	**RMSMAN**	**GARDS_RELEVANT_NUCLIDES**	**TABLE**	Contains the list of nuclides determined to be relevant during the identification of radionuclide events. This table includes the categorization list for particulate samples (CLPS) and the relevant radioxenon isotopes
GARDS_RESOLUTION_CAL	**TABLE**		**GARDS_RESOLUTION_CAL**	**TABLE**	Contains resolution calibration coefficients calculated during analysis
GARDS_RESOLUTION_CAL_COV	**TABLE**		**GARDS_RESOLUTION_CAL_COV**	**TABLE**	The table contains the covariance table for each coefficient of the resolution calibration from a given sample_id

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_RESOLUTION_CAL_ORIG	TABLE		GARDS_RESOLUTION_CAL_ORIG	TABLE	Contains the original resolution calibration coefficients calculated during analysis
GARDS_RESOLUTION_PAIRS	TABLE		GARDS_RESOLUTION_PAIRS	TABLE	Contains resolution calibration pairs information associated with spectral PHD
GARDS_RESOLUTION_PAIRS_ORIG	TABLE		GARDS_RESOLUTION_PAIRS_ORIG	TABLE	Contains the original resolution calibration pairs information associated with spectral PHD
GARDS_RLR	TABLE	RMSAUTO	GARDS_RLR	SYNONYM	Contains all data related to a Radionuclide Laboratory Report (RLR) message
GARDS_RLR_CONCLUSIONS	TABLE	RMSAUTO	GARDS_RLR_CONCLUSIONS	SYNONYM	Contains interactive analysis comments for RLR samples
GARDS_RLR_OBJECTIVE	TABLE	RMSAUTO	GARDS_RLR_OBJECTIVE	SYNONYM	Contains analyze purpose, special instructions and authorized tests for RLR samples
GARDS_RLR_RATIOS	TABLE	RMSAUTO	GARDS_RLR_RATIOS	SYNONYM	Contains activity ratios for RLR samples
GARDS_RLR_RESULTS	TABLE	RMSAUTO	GARDS_RLR_RESULTS	SYNONYM	Contains analysis results for RLR samples
GARDS_RLR_SEQ	SYNONYM	RMSMAN	GARDS_RLR_SEQ	SEQUENCE	Sequence that creates autonumbers for RLR IDs
GARDS_RLR_SSREB	TABLE	RMSAUTO	GARDS_RLR_SSREB	SYNONYM	Contains RLR and related sample IDs
GARDS_ROI_CHANNELS	TABLE		GARDS_ROI_CHANNELS	TABLE	Contains the ROI boundaries in channel units that are calculated in *rms_xanalyze*
GARDS_ROI_CONCS	TABLE		GARDS_ROI_CONCS	TABLE	Contains information regarding the concentration of each identified nuclide that is calculated in r*ms_xanalyze.* Seems not to be used since rms_xanalyze has been replaced by bg_analyze
GARDS_ROI_COUNTS	TABLE		GARDS_ROI_COUNTS	TABLE	Contains information regarding the counts that are calculated in r*ms_xanalyze* for each ROI
GARDS_ROI_LIB	SYNONYM	RMSMAN	GARDS_ROI_LIB	TABLE	Relates each ROI to a specific nuclide. The table also contains nuclide properties used in nuclide quantification
GARDS_ROI_LIMITS	TABLE		GARDS_ROI_LIMITS	TABLE	Contains ROI boundaries in energy units as specified in the PHD file
GARDS_ROLE_ID_SEQ	SYNONYM	RMSMAN	GARDS_ROLE_ID_SEQ	SEQUENCE	Sequence that creates autonumbers for ROLE IDs
GARDS_ROLES	SYNONYM	RMSMAN	GARDS_ROLES	TABLE	Contains the list of roles that can be manipulated by the MAR tools
GARDS_ROLES_PERMISSIONS	SYNONYM	RMSMAN	GARDS_ROLES_PERMISSIONS	TABLE	Contains a mapping of which permissions are assigned to which roles

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_SAMPLE_AUX	TABLE	RMSMAN	GARDS_SAMPLE_AUX	SYNONYM	Contains auxiliary information related to raw sample data
GARDS_SAMPLE_AUX_TEMP	TABLE		GARDS_SAMPLE_AUX_TEMP	TABLE	Table contains same fields as GARDS_SAMPLE_AUX and has just one entry from Feb 2005. Table is probably not longer in use
GARDS_SAMPLE_CAT	SYNONYM	RMSMAN	GARDS_SAMPLE_CAT	TABLE	Contains the most recent categorization values from either automatic or manual processing
GARDS_SAMPLE_CERT	TABLE	RMSAUTO	GARDS_SAMPLE_CERT	SYNONYM	Contains overview information regarding certificate data found in pulse height data messages
GARDS_SAMPLE_CERT_LINES	TABLE	RMSAUTO	GARDS_SAMPLE_CERT_LINES	SYNONYM	Contains nuclide information regarding certificate data found in pulse height data messages
GARDS_SAMPLE_DATA	TABLE		GARDS_SAMPLE_DATA	TABLE	Contains header data from PHD messages
GARDS_SAMPLE_DESCRIPTION	TABLE	RMSAUTO	GARDS_SAMPLE_DESCRIPTION	SYNONYM	Contains description and comment text as specified in a PHD file
GARDS_SAMPLE_FLAGS	TABLE		GARDS_SAMPLE_FLAGS	TABLE	Contains the result of the *rms_DBflags* analysis for each sample
GARDS_SAMPLE_ID_SEQ	SYNONYM	RMSMAN	GARDS_SAMPLE_ID_SEQ	SEQUENCE	Sequence that creates autonumbers for sample IDs
GARDS_SAMPLE_PROC_PARAMS	TABLE		GARDS_SAMPLE_PROC_PARAMS	TABLE	Contains specific processing parameters to be used for a particular spectral analysis; the table contains a combination of values from GARDS_PROC_PARAMS_TEMPLATE and values entered at the command line
GARDS_SAMPLE_RATIOS	TABLE		GARDS_SAMPLE_RATIOS	TABLE	Contains the amount of overlap between each ROI as specified in the PHD file
GARDS_SAMPLE_STATUS	TABLE		GARDS_SAMPLE_STATUS	TABLE	Contains spectral processing historical data
GARDS_SAMPLE_UPDATE_PARAMS	TABLE		GARDS_SAMPLE_UPDATE_PARAMS	TABLE	Holds the actual parameters used for energy calibration updating during analysis. This table may contain a combination of defaults from GARDS_UPDATE_PARAMS_TEMPLATE and values from the command line

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_SAMPLE_XE_PROC_PARAMS	TABLE		GARDS_SAMPLE_XE_PROC_PARAMS	TABLE	Contains specific processing parameters to be used for a particular spectral analysis
GARDS_SOH_CHAR_DATA	TABLE	RMSAUTO	GARDS_SOH_CHAR_DATA	SYNONYM	Contains all state of health (SOH) character data
GARDS_SOH_CODE	SYNONYM	RMSMAN	GARDS_SOH_CODE	TABLE	Contains all state of health (SOH) parameter names and their codes
GARDS_SOH_HEADER	TABLE	RMSAUTO	GARDS_SOH_HEADER	SYNONYM	Contains the header information for all SOH messages
GARDS_SOH_ID_SEQ	SYNONYM	RMSMAN	GARDS_SOH_ID_SEQ	SEQUENCE	Sequence that creates autonumbers for soh IDs
GARDS_SOH_NUM_DATA	TABLE	RMSAUTO	GARDS_SOH_NUM_DATA	SYNONYM	Contains all state of health (SOH) numerical data
GARDS_SOH_SENSOR_DATA	TABLE	RMSAUTO	GARDS_SOH_SENSOR_DATA	SYNONYM	Contains the data reported in the #ProcessSensors data block in noble gas RMSSOH messages
GARDS_SPECTRUM	TABLE	RMSAUTO	GARDS_SPECTRUM	SYNONYM	Contains information regarding the spectral data information reported in the PHD file
GARDS_STADET	SYNONYM	RMSMAN	GARDS_STADET	TABLE	Contains a complete list of all station/detector identifier code combinations. The entries in this table are generated via a trigger on the GARDS_SAMPLE_DATA table
GARDS_STATION_ASSIGNMENTS	SYNONYM	RMSMAN	GARDS_STATION_ASSIGNMENTS	TABLE	Contains a mapping of which stations are assigned to which users
GARDS_STATION_ID	SYNONYM	RMSMAN	GARDS_STATION_ID	SEQUENCE	Sequence that creates autonumbers for STATION IDs
GARDS_STATIONS	SYNONYM	RMSMAN	GARDS_STATIONS	TABLE	Contains a station overview and station characteristics
GARDS_STATIONS_SCHEDULE	SYNONYM	RMSMAN	GARDS_STATIONS_SCHEDULE	TABLE	Contains schedule descriptions for stations reporting to a data centre
GARDS_STATUS_HISTORY	SYNONYM	RMSMAN	GARDS_STATUS_HISTORY	TABLE	Contains a list of changes in GARDS_SAMPLE_STATIONS.*status* for reviewed samples
GARDS_TOTAL_EFFIC	TABLE		GARDS_TOTAL_EFFIC	TABLE	Contains detector total efficiency data as specified in the PHD file

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
GARDS_TRENDVUE	TABLE		GARDS_TRENDVUE	TABLE	Contains data produced by the *Trendvue* application and is purged at the end of each section
GARDS_UPDATE_PARAMS_TEMPLATE	SYNONYM	RMSMAN	GARDS_UPDATE_PARAMS_TEMPLATE	TABLE	Holds the default parameters used for energy calibration updating during analysis. These values can be overridden at the command line
GARDS_UPDATE_REFLINES	SYNONYM	RMSMAN	GARDS_UPDATE_REFLINES	TABLE	Contains the energies of the reference lines used in the ECR/RER updating functions
GARDS_USER_COMMENTS	TABLE		GARDS_USER_COMMENTS	TABLE	Contains text associated with user-defined comments
GARDS_USER_ID_SEQ	SYNONYM	RMSMAN	GARDS_USER_ID_SEQ	SEQUENCE	Sequence that creates autonumbers for USER IDs
GARDS_USERENV	SYNONYM	RMSMAN	GARDS_USERENV	TABLE	Contains configurable environment variables used by the automatic processing software
GARDS_USERS	SYNONYM	RMSMAN	GARDS_USERS	TABLE	Contains a list of login names for radionuclide related personnel
GARDS_USERS_ROLES	SYNONYM	RMSMAN	GARDS_USERS_ROLES	TABLE	Contains a mapping of which roles are assigned to which user
GARDS_WRITE_SAMPLE_CAT	SYNONYM		GARDS_WRITE_SAMPLE_CAT	SYNONYM	Synonym for GARDS_ SAMPLE_CAT
GARDS_XE_NUCL_LIB	SYNONYM	RMSMAN	GARDS_XE_NUCL_LIB	TABLE	Contains nuclide library information used by the *Inspectra* when displaying nuclide data for gamma spectroscopy noble gas stations
GARDS_XE_NUCL_LINES_LIB	SYNONYM	RMSMAN	GARDS_XE_NUCL_LINES_LIB	TABLE	Contains nuclide library information for specific lines of a nuclide and is used by the *Inspectra* when displaying nuclide data for gamma spectroscopy noble gas stations
GARDS_XE_PROC_PARAMS_TEMPLATE	SYNONYM	RMSMAN	GARDS_XE_PROC_PARAMS_TEMPLATE	TABLE	Contains information used for nuclide identification. Probably not longer in use. Last entry 2004. Table was associated to *rms_xanalyze*
INTVL_SEQ	SEQUENCE				Sequence that creates autonumbers for INTERVAL IDs
			JAVA$CLASS$MD5$TABLE	TABLE	
			JAVA$OPTIONS	TABLE	
LASTID	SYNONYM	IDCX	LASTID	SYNONYM	

RMSAUTO			RMSMAN		DESCRIPTION
TABLE_NAME	TYPE	SYNONYM FROM	TABLE_NAME	TYPE	
MET_SEQ	**SEQUENCE**				Sequence that creates autonumbers for MET IDs
MSGDISC	**SYNONYM**	**IDCX**	**MSGDISC**	**SYNONYM**	Synonym for MSGDISC (IDCXDB)
SOH_CHAR_SEQ	**SEQUENCE**				Sequence that creates autonumbers for SOH CHAR IDs
SOH_NUM_SEQ	**SEQUENCE**				Sequence that creates autonumbers for SOH NUM IDs
SOH_SENSOR_SEQ	**SEQUENCE**				Sequence that creates autonumbers for SOH SENSOR IDs
UPDATECATRECORD	**SYNONYM**		**UPDATECATRECORD**	**PROCEDURE**	UPDATECATRECORD (rmsman)

TERMINOLOGY

Glossary

ClearCase branch

A set of sequentially numbered versions within the version tree of a *ClearCase* object. All version trees have a `main` branch and can have additional branches with unique branch names.

ClearCase label

A label that is applied to an individual version of a *ClearCase* object. Various labels are used for the Development LAN to define versions, which are part of a software baseline or patch as well as to define versions that are currently installed in the runtime system.

ClearCase
version tree

The set of all versions of a *ClearCase* object (i.e. a directory or file), which are stored in *ClearCase*. The version tree starts with version `0` on the `main` branch and can have additional branches each holding sequentially numbered versions. At the IDC the *ClearCase* version tree holds the history of files and directories on all IDC LANs as well as all development versions. A *ClearCase* version string consisting of branch names and the version number uniquely defines each version in the version tree.

ClearCase view

A configured set of rules in *ClearCase* to select unique versions from all version trees of objects in a given VOB. Individual views are identified by view names. The views used for the Development LAN select versions based on their branch names or on specific labels.

ClearCase VOB

A logical area in *ClearCase* to manage a set of *ClearCase* objects including all their versions. The IDC *ClearCase* VOBs contain high-level system directories including all their subdirectories, files and links. Each *ClearCase* VOB corresponds to a directory on the runtime system.

cron job

Cron is a utility of the UNIX/Linux operating systems to automatically run individual commands, scripts or applications at configurable regular times. All IDC scripts and applications, which are not controlled by the DACS or by other scripts or applications, are controlled by *cron*. Individual *cron* jobs can be configured in the local `crontab` file for individual authorised users on each machine.

data

Files, which are written by the IDC processing system, are considered data if they are re-used by the system for further processing. Otherwise they are considered products. This categorisation is used in the Development LAN directory structure. Data and product files are not under *ClearCase* version control.

Development
LAN

The computer hardware and infrastructure used to integrate and test software and configuration changes as well as new software at the IDC before promoting the changes to the Testbed LAN for operational testing. Physically separate from the Operations and Testbed LANs and under less rigorous configuration control.

`devlan` branch — The `devlan_view` is configured to automatically create a new version on the `devlan` branch if a directory or file is modified and checked-in in the `devlan_view`. The latest sequential version on the `devlan` branch is always the latest version that has been created under the `devlan_view`. If the version tree of a *ClearCase* object does not have a `devlan` branch this *ClearCase* object has never been modified and checked-in in the `devlan_view`.

`devlan_rn_view` — The `devlan_rn_view` is the *ClearCase* view to be used to modify and check-in versions of Radionuclide software-related *ClearCase* objects for the Development LAN. It selects versions based on the same branch names as the standard `devlan_view`. However, the `devlan_rn_view` enables users to check-out and check-in versions in the Radionuclide software *ClearCase* VOB.

`devlan_solaris_view` — The `devlan_solaris_view` is exclusively used to build Linux compatible software on Solaris platforms for backwards compatibility reasons. It is configured to check-in versions on the `devlan_solaris` branch. Since source code is uniformly maintained on the `devlan_branch` for both Solaris and Linux platforms the view will select the latest source code version on the `devlan` branch if this branch exists in the version tree of the *ClearCase* object. If no `devlan` branch exists for a particular object the view will use the same selection rules as the standard `devlan_view`. Only *ClearCase* objects which are installed via the Solaris build procedure in this view are supposed to have a `devlan_solaris` branch.

`devlan_view` — The `devlan_view` is the standard *ClearCase* view to be used to modify and check-in versions of *ClearCase* objects for the Development LAN. It is configured to select versions based on their branch names. It will select the latest version on the `devlan` branch if this branch exists in the version tree of the *ClearCase* object. It will select the latest version on the `R3_tst` branch if the `devlan` branch does not exist. If neither a `devlan` branch nor an `R3_tst` branch exist in a given version tree the `devlan_view` will select the latest version of the *ClearCase* object on the `main` branch. This latter case occurs for files or directories that have never been modified either on the Development LAN or on the Testbed.

`DEV_LAN` label — The `DEV_LAN` label defines *ClearCase* versions, which are currently installed in the Development LAN runtime system. It can only be applied to a single unique version in the version tree of each *ClearCase* object. The `DEV_LAN` label is manually applied when a new version is promoted from *ClearCase* to the Development LAN runtime system.

`DEV_LAN` view — The `DEV_LAN` view is configured to select only versions, which have the `DEV_LAN` label applied. If the `DEV_LAN` labels are correctly set the `DEV_LAN` view will show exactly the same versions as the Development LAN runtime system directories.

Operations LAN — The computer hardware and infrastructure used for operational data processing at the IDC.

pipeline — Well-defined processing sequence during automated processing, where the same data time interval is successively subjected to processing by several applications.

products Files, which are written by the IDC processing system, are considered products if they are not further processed and are made available to (external) users. Otherwise they are considered data. This categorisation is used in the Development LAN directory structure. Data and product files are not under *ClearCase* version control.

runtime system The directories and files, which are mounted on the machines of an IDC LAN and which are used to operate the IDC processing software. All directories and files in the runtime system, which are under version control, have a counter part in *ClearCase*. The Development LAN runtime versions correspond to *ClearCase* versions that have the `DEV_LAN` label applied. Data files, products and log files are not under version control and exist only on the runtime system.

Testbed LAN The computer hardware and infrastructure used to test new stations and software changes in an operational environment before installation in the Operations LAN. A close copy but physically separate and independent from the Operations LAN.

Abbreviations

ATM	Atmospheric Transport Modelling
CCB	Configuration Control Board
CDS	Continuous Data Subsystem
CIN	Change Implementation Note
CSCI	Computer Software Configuration Item
CRP	Change Request Proposal
CTBTO	Comprehensive Nuclear-Test-Ban Treaty Organisation
DACS	Distributed Application Control Subsystem
DCR	Development LAN Change Request
Dev LAN	Development LAN
ECMWF	European Centre for Medium-Range Weather Forecasts
FOR	Field of Regard
IDC	International Data Centre
IMS	International Monitoring System
LAN	Local Area Network
LEB	Late Event Bulletin
NCEP	U.S. National Centres for Environmental Prediction
NDC	National Data Centre
PHD	Pulse Height Data
PSR	Possible Source Region
PTS	Provisional Technical Secretariat
REB	Reviewed Event Bulletin
RN	Radio-Nuclide technology
RRR	Reviewed Radionuclide Report
SEL	Standard Event List
SHI	Seismic, Hydro-acoustic and Infrasound technologies
SI	Software Integration unit
SRS	Standardized Source-Receptor Sensitivity
VOB	Versioned Object Base

REFERENCES

[IDC/DEVLAN/OG-SHI] Development LAN Operator's Guide -- SHI

[IDC/DEVLAN/OG-ATM] Development LAN Operator's Guide -- ATM

[CIN-01462] Installation of new IMS Radionuclide Stations on the Testbed.

[IDC-CCB] IDC CCB Procedures, http://idc030.idc.ctbto.org:8000/Docs/CCB-Documentation.html.

[IDC-OM] IDC Operational Manual, CTBT/WGB/TL-17/2/Rev.1, June 2002.

[IDC-SDO] IDC Software Documentation Orientation Booklet, Version 0.2, IDC/WI, March 2002.

[IDC3.4.1] Formats and Protocols for Messages, Revision 6.

[IDC5.1.1] Database Schema, Revision 5.

[IDC5.1.3] Configuration of PIDC Databases, Revision 1.

[IDC5.2.2] IDC Processing of Radionuclide Data, Revision 3.

[IDC6.2.4] Configuration of IDC Processing Data Files, Revision 1.

[IDC6.5.10] Radionuclide Software User Manual, Revision 4.

[IDC6.5.19] Message Subsystem Software User Manual.

[IDC6.5.21] Subscription Subsystem Software User Manual.

[IDC7.4.2] Message Subsystem.

[IDC/autoSaint/SUG] autoSaint Software User Guide