PrimeCode® CRW

Custom Report Writer

User Manual

CRW_D20_2.0.3

for PrimeCode 3.5.x with RMS D20.xx

Document History

Edition	Release	Date
First Edition	CRW 1.0.0	May 2001
Second Edition	CRW 1.1.0	December 2001
Third Edition	CRW 1.1.1	March 2002
Fourth Edition	CRW 2.0.0	September 2002
Fifth Edition	CRW 2.0.0	May 2003
Sixth Edition	CRW D20 2.0.1	September 2003
Seventh Edition	CRW D20 2.0.1	May 2006
Eighth Edition	CRW D20 2.0.3	February 2009

CRWX D9023 D20.07 - D20.15

5955 Airport Road, Suite 200,
Mississauga, Ontario, Canada. L4V 1R9

(905) 677-6666 (905) 677-6671 sales@emperex.com

www.emperex.com

Table of Contents

Preface

Configuration

Updating the Database

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

Preface

Introduction

PrimeCode CRW - Custom Report Writer is a PC application designed to complement PrimeCode, a software configuration management tool from Emperex. PrimeCode is a powerful tool that can be used to control every facet of the program development life cycle but has limited reporting capabilities. Much useful information is stored in the HP NonStop Enscribe database used by PrimeCode but is inaccessible by more common PC tools.

PrimeCode CRW will facilitate access to this information by extracting data from the Enscribe database to produce a SQL database on a Windows PC. On NT and Windows 2000 systems, a System DSN can be setup to make this data available to any computer on the network via ODBC.

Access to the details from this source could provide important statistics for monitoring, trend analysis and usage pattern determination.

Several built-in report generators are included with PrimeCode CRW and using SQL queries or tools such as Microsoft Access or Crystal Reports, users can design and run their own custom reports.

Purpose

This User Guide details the components of the PC interface for PrimeCode CRW and its operation for updating a database which is analogous to the RMS Catalog. An overview of the data in the database is included to assist in the design of custom reports.

Who Should Read This Manual

This manual is intended for users of PrimeCode CRW and/or the database it creates. In this guide you will find procedures for updating a SQL database and field descriptions for the data which is available to query for reporting purposes.

Related Documentation

- PrimeCode CRW Installation/Configuration and Administration Guide
- PrimeCode CRW On-line Help

- PrimeCode Installation Guide
- PrimeCode User Manual

Customer Support

We offer two types of support. Send us an e-mail or call the numbers below.

E-mail: (support@emperex.com)

Phone:
Regular Hours (Monday - Friday, 9 a.m. - 6 p.m. Eastern Standard time) at (905) 677-6666

After Hours Support: (905) 667-6666.

System Requirements

PC Requirements

Minimum

Operating System	***Windows 95***
Memory	32 MB
Free hard disk space	
for application	5 MB
for database	1.5 MB / 1000 components

Recommended

Operating System	***Windows 95, 98, 2000, XP or Windows NT***
Memory	
for 700,000 components	32 MB
for 7,000,000 components	64 MB
for 32,000,000 components	96 MB
for 51,000,000 components	128 MB
Free hard disk space	
for application	5 MB
for database	1.5 MB / 1000 components

HP NonStop/Guardian Requirements

A working FTP server will be required to upload the CRWD20xx file from the CD to the HP NonStop and to download the ASCII files from the HP NonStop to the ASCII file directory on the PC.

There are no other special requirements here that would not be already satisfied in order to run PrimeCode.

System Overview

An analogous database will be created on a Windows PC comprised of data from the PrimeCode Enscribe database. This local database can then be used to generate reports. A two-tier system involving both the HP NonStop and a Windows PC is used.

The first step in this process will be the creation of several ASCII files containing the extracted data from the Enscribe database. A procedure for updating these text files at regular intervals can be established using a batch scheduler to launch a program. The ASCII files will be stored in a $volume.subvolume location on the HP NonStop designated by the user during installation of the HP NonStop components of this software.

Guardian Username and Password combinations are required as communication with the HP NonStop is involved and users must logon to the HP.

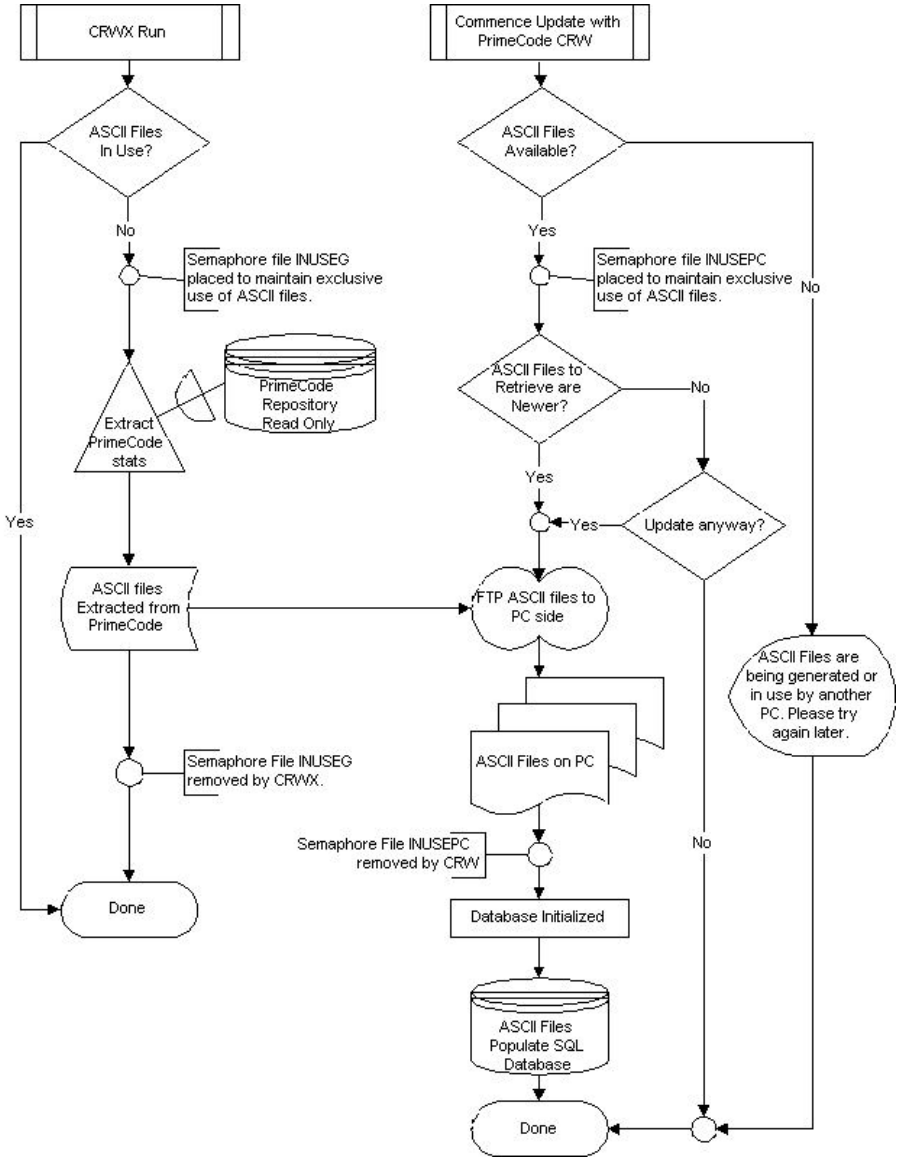
Profiles enable users to specify both the input and the location for the output created when they use that profile, thus maintaining private copies of the SQL database created on the PC platform. Updates of these personal databases can be initiated when appropriate to the needs of the individual user who implements a profile.

With Username/Password verification and login to the HP, PrimeCode CRW will initiate an FTP connection between the PC and the HP machine to download the ASCII files from the NonStop. A local SQL database will be updated with the new data.

PrimeCode CRW's database includes several built-in reports which can be run right away using MS Access.

PrimeCode CRW Process Flow

Illustration shows the HP NonStop process on the left and the PC application on the right. With regards to the PrimeCode repository and the ASCII files which are extracted from it, all processes are Read Only.



Installing the PC Application

If any previous versions of PrimeCode CRW have been installed it is necessary to first un-install before continuing. See Un-Installing below.

Running the .msi Install File

Insert the PrimeCode CRW CD into the CD ROM drive.

An auto run program may display a menu of choices.

Select **Install PrimeCode CRW**
If you are not configured for AutoRun use the **Start/Run** procedure below.

Running install file from the Auto Run Menu requires that you select to run the installation from its current location.

START RUN

1. Click the ***Start*** button on the Windows Task Bar and select ***Run***.
2. Navigate, using the ***Browse*** button to the Setup.exe program :
   D:\PrimeCode_FCS_x.x.x\CRW\crw_d20_2.0.1.msi
   (Where D: is assumed to be the CD ROM drive)
   OR
   To display the Auto-Run Menu, run this line instead:
   D:\PrimeCode_FCS_x.x.x\default.htm
   (The Browse dialog will initially display only files of type: program. Switch this to All Files to see the default.htm file.)
3. Click ***Open***.
4. Click ***OK***.

PrimeCode CRW will prompt you to close all running applications.

PrimeCode CRW will be installed by default to

C:\ProgramFiles\Custom Report Writer

To change this, click on the ***Change Directory*** button and then use the Drive/File Combo boxes in the dialog to specify a different location.

Continued...

Accept the suggested program group name, add Custom Report Writer to an existing group or name a new group which will be created.

> If PrimeCode CRW attempts to load any file that is not newer than a current version of that file found on your PC, Windows will prompt you to keep or overwrite the file you have. It is usually safer to answer "Yes" and keep your newer version file.

A successful install will be acknowledged by the message "Custom Report Writer has been sucessfully installed."

Un-Installing the PC Application

To thoroughly uninstall Custom Report Writer, run the crw_d20_2.0.1.msi file again or use the Add/Remove programs utility in Control Panel.

> If you have created a local database in the same directory, Un-install will be unable to completely remove all the contents from the Custom Report Writer directory. Manually delete files not installed by PrimeCode CRW.

PrimeCode CRW maintains the parameters of your established Profiles in the Windows Registry. When you uninstall PrimeCode CRW, these registry entries are preserved so that new installations will have knowledge of Profiles you have created. If you wish to uninstall PrimeCode CRW and completely remove all traces of it you must manually remove these registry enties.

Removing Custom Report Writer from Windows

1. Click the ***Start*** button on the Windows Task Bar.
2. Select Settings\Control Panel.
3. Double Click the "Add/Remove Programs" icon.
4. Scroll through the programs listed and select "Custom Report Writer".
5. Click the ***Add/Remove*** button.
6. Click ***Yes*** to confirm your choice or ***No*** to cancel.
7. Click ***Yes*** to keep any shared files.
8. Click the ***OK*** button to finish.
9. Click ***OK*** to close the Add/Remove Dialog.
10. Close the Control Panel.

Configuration

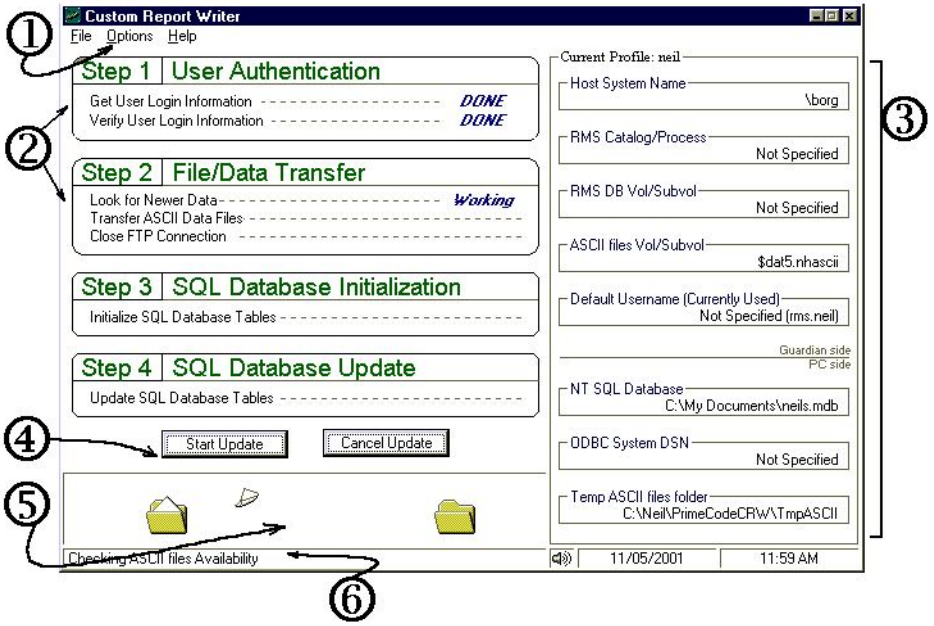
Main Window Features

The PrimeCode CRW Main Window has two major components.

The left side of the window illustrates the various stages in the process of updating the SQL database. Confirmation of each stage is given as it is completed.

The right side panel illustrates the parameters of the current profile in use. The Default Profile Parameters are shown for your visual confirmation.

The extended view of the Main window, showing the profile parameters can be toggled on and off by selecting Expanded View from the Options menu or by pressing Ctrl+E.



1 - Main Menu

File, Options and Help menus are available on the Main Menu.

File

Exit Alt+F4 — Closes Custom Report Writer.

Options

Application Settings Ctrl+S — Opens a dialog for adjusting profile parameters or creating/selecting a new profile. This dialog also provides access to customizable features in the GUI.

Expanded View Ctrl+E — The profile parameters display can be toggled on and off by checking this option.

Help

Contents F1 — Launch On-line Help.

Visit the Emperex Website Ctrl+D — This option will launch the default browser and go to the Emperex Corporation web site.

Visit Product Website Ctrl+P — Visit Product Website. This option will launch the default browser and go to the product website. Newer versions of help will be available here as they are created.

About Ctrl+A — Details of your version of Custom Report Writer.

The About Box provides a button for displaying useful system information and links to our websites.

2 - Step Confirmation Panel

The four steps in the update process are illustrated on the left side of the main window and confirmation of their completion is displayed as each step is finished.

3 - Profile Parameters Panel

The current profile parameters are displayed on the right side of the main form (in Expanded View mode). This is visual confirmation for the user that the correct database is being updated and being accessed by intended user.

4 - Buttons

Two buttons, ***Start Update*** and ***Cancel Update*** are alternatively greyed out when running the application. Click on ***Start Update*** to begin the SQL database update.

5 - Progress Area

The progress area will indicate progress for various stages of the updating process with file transfer animations and progress bars to confirm that the application is running.

6 - Status Bar

Short descriptions of the current action are displayed on the status bar.

An icon indicating the status of the sound for this application is displayed. Double clicking this icon will toggle the sounds on and off.

The status bar displays the current date and time.

Profiles

What Are Profiles?

PrimeCode CRW can be launched but cannot perform its main objective of updating your SQL database without a valid User Profile.

A User Profile governs the file transfer process from the HP NonStop computer to your PC, specifying which ASCII files to bring over and where to put them. It also controls which local SQL database will be created/updated to reflect the data collected from PrimeCode or RMS.

Why Use Profiles?

Profiles allow a user to specify all of the parameters of a SQL database update by referring to just the profile name. More importantly, profiles allow individuals to maintain separate copies of a local database and ensure that only the appropriate database gets updated and at intervals that satisfy their requirements.

Example : If the ASCII files are prepared frequently, one analyst may wish to work with the newest data that is contained in these files for an up to date picture of the statistics and will update frequently. Another however may want to work with data that was retrieved at a certain date and frozen to reflect, for example, the statistics for the year ending. Anyone updating the same database he/she is evaluating will refresh the statistics to reflect the condition at the latest time the Guardian ASCII files were generated, in effect losing the view of the database at year end.

Profiles contain essential information about how an update will take place. In fact the profile controls what gets updated. A profile must exist for an update to be executed.

Creating or maintaining profiles is done through the PrimeCode CRW PC Interface.

To create a profile:

1 Launch the PC program Custom Report Writer.

2. Select *Profile Management* from the *Options* dialog.

   [From the Main Menu ; Options \ Application Settings or Ctrl+S]

   First Profile

   If this is the first profile you have created all the fields are blank. Mandatory fields required in the creation of a profile are indicated by an asterisk. Supply all mandatory fields and any others you want recorded in the profile.

   New Profiles

   If any profiles have been created previously, one of them will be currently loaded as the Default Profile. You can simply type over some or all of the fields with the exception of NT SQL Database and Temp ASCII Files to clone and existing profile by modifying and renaming it. Creating and modifying the NT SQL Database field is described on page 11.

3. When finished creating, modifying or adding a profile,

   Click ***Save As New Profile***.

DO NOT click the OK button. This will close the Options dialog and save the modifications to the current profile, failing to create a new one.

Field sensitive help is available for each of the fields on the form.

PROFILE COMPONENTS

The following pages describe each of the fields on the *Profile Management* dialog. Field sensitive help is available for each of these in the on-line help.

NAME

***Profile Name**

The profile name is used to identify a set of parameters which control the way a database update is carried out. You can invoke an entire saved environment for your SQL database update by loading a single profile.

Naming Rules: Profile names may contain a maximum of 16 characters and may include spaces.

GUARDIAN OS SIDE COMPONENTS

***Host System Name**

The host system name of the HP NonStop computer you will access must be provided. Alternatively, the IP address of the HP NonStop can be used.

RMS Process Name

This field is for your reference only. It will not affect the way PrimeCode CRW will work.

The process RMSMON, the monitor process for RMS which underlies the PrimeCode software, resides with the Enscribe database on the HP NonStop machine. For your visual confirmation you can record here, the RMS Catalog's Process name (i.e.. $KRMP) to confirm which Catalog or database you are going to be associating with. If you do not know where your catalog is you can start RMSCOM by typing the following at a TACL prompt.

```
<Volume>.<Subvolume>.RMSCOM <$MonitorName> [Enter]
```

The response should indicate your catalog location on a line similar to the following;

CURRENT CATALOG IS \BORG.$DAT5.KRMS

RMS DB Vol/Subvol

This field is for your reference only. It will not affect the way PrimeCode CRW will work.

For your visual confirmation you can record here the Volume and Subvolume for the location of your RMS database.

*ASCII Files Vol/Subvol

A program residing on the HP NonStop computer will regularly produce a set of ASCII files which contain the relevant content of the RMS Enscribe database. Specify here, where on the HP these files have been saved.

In the Guardian operating system, the format of a Volume**.**Subvolume location is:

$volume**.**subvolume

Default Username

This field is optional.

For normal security, a Login for the HP NonStop is required when you initiate an update process. Any valid Guardian Username can be given at this time but you can include your Username, for convenience among the saved parameters in your profile so that it is automatically filled for you.

Example: The User Profile specifies the Default Username “RMS.NEIL” and this field is auto-filled when you begin your update.

***NT SQL Database**

The name of your SQL database is mandatory. This database is the destination for the information contained in the files that PrimeCode CRW will download from the HP NonStop computer.

This field is not edited directly but supplied using the ***Browse*** button to its right.

To use an Existing Database created by PrimeCode CRW

1. Click the ***Browse*** button.
2. Navigate to the location of an existing database and select it.
3. Click ***Open***.

To create a New Database

1. Click the ***Browse*** button.

   The Open File dialog is presented.

2. Navigate to the folder where you want the new database to reside.

> It is recommended that the database be located in any folder other than that where PrimeCode CRW was installed. This will simplify your uninstall procedure should you ever need to uninstall (Uninstalling is required when updating your version of PrimeCode CRW).

3. Type the name of the new database in the *Filename* field and click ***Open***.

   PrimeCode CRW will prompt you for permission to create the new database.

4. Click ***Yes***.

ODBC System DSN

The ODBC System Data Source Name is saved in your profile for reference. PrimeCode CRW does not respond any differently depending on the input to this field. It is basically a reminder to any user who has created a database, what alias name points to the correct driver for accessing the database you created. An ODBC driver contains information about how to connect to a certain type of database. A system ODBC DSN, unlike a user DSN, is available to all users on the network.

Future versions of PrimeCode CRW will generate and/or associate System DSNs with generated SQL databases.

*Temp ASCII files folder

Specify the location on your local filling system where the ASCII files retrieved from the HP NonStop will be stored. This is a mandatory field in the collection of data that is used to update your SQL database.

Use the ***Browse*** button to navigate to the desired folder.

Loading Profiles

Loading another profile changes the location from where PrimeCode CRW will retrieve the ASCII files and the location of the resulting database.

To load a profile, making it the default;

1. From the *Main Menu*,
   Select *Options \ Application Settings or Ctrl+S*
2. Click the ***Load*** button next to the profile name.
3. Select one of the existing profiles and click ***OK***.

Modifying Profiles

To modify an existing profile's name or parameters you must load it, making it the current default profile.

To load a new default profile see Loading Profiles above.

Make appropriate changes and click ***Update Profile*** or ***OK***.

Clicking Update Profile will save your changes but not close the Options dialog. You can reload any other profile or make other modifications to the current one.

Clicking OK will save your changes and close the Options dialog. The last loaded profile becomes the Default Profile.

Deleting Profiles

The current profile will be deleted and the current default will be cleared. You will be prompted to specify a new default profile.

1. Click the ***Delete Profile*** button.
2. Confirm the deletion.
3. Select a replacement Default Profile from the *Profiles List Box*.

GUI Options

The GUI (graphical user interface) of this program can be modified slightly to suite a preference. The modifiable components of this interface are the Splash Screen and the Sounds.

Splash Screen

You may control whether or not the splash screen will display on start-up and if so, for how long.

You may toggle on and off, the splash screen display on start-up with the check box.

If the "Show Splash Screen on Start-up" check box is checked you may select from the dropdown list a choice for the number of seconds the splash screen will display.

Sounds

Custom Report Writer emits sounds on two events:

- Each time a step in the update process completes.
- When an update is finished.

To change the .wav file which is played for either of these events, click the ***Browse*** button to the right of the field and navigate to the .wav file of your choice in the Open dialog and select it.

Check the ***Use Sounds*** checkbox to enable or disable sounds. This feature can be controlled by double clicking the sounds icon on the status bar as well.

You can test your chosen sound by clicking on the filename after selecting it.

Updating the Database

Start

To begin Custom Report Writer's generation of a SQL database...

1. Click on the ***Start Update*** button.

Logging In

Once you have created at least one valid profile you can launch the Update Database function of Custom Report Writer. If you do not have a profile set up refer to the Configure PC Application on page 3.

To continue the update you must login to the HP machine to access the extracted ASCII files produced by the HP component of this software. The User Authentication Window will display the profile being used and the host system to be connected to. Users are prompted for Username and Password.

Using Profile	Indicates current profile used.
Host System	The node name of the Guardian system where your PrimeCode Catalog resides. Auto filled by information in your profile.
Username	Your Guardian Username (group.user). For ease of use, you can store this parameter in your profile and have this field auto filled.
Password	Your password is required to log in to the host system.

To Log in:

1. Provide a Username and Password.
2. Click the ***OK*** button

The update process, governed by the profile you have loaded will resume. Watch for confirmation of each step as it is completed on the main form. When Custom Report Writer has finished, if there are no other databases to update, close the program and launch whichever reporting tool you intend to work with.

The Database

The SQL database is MS Access 2000 compatible.

APPENDIX A

Database Tables

The following tables within the database created by Custom Report Writer correspond to the individual tables in the RMS Enscribe database. A full field description for each table is given in Appendix D.

Table Name	Description
ARCHENT	The Archive contents. This contains all Data files and Objects which have been submitted as well as all compiled Objects.
AUXENT	Auxiliary Attributes, many of which are user defined qualities of a Component.
COMPDREL	Rules of compiled relationships - relationships defined between Source and Object Components which have been involved in a compile. This relationship is a permanent link between two Components. A Rule Id has been associated with this Rule superseding the Rule name.
COMPENT	Component Info.
COMPREL	Rules of compile relationships - relationships defined between Source and Object Components which have not yet been used in a compile. A Rule name identifies the Rule and it can be edited or renamed at this point.
DEPREL	Dependencies for each Version. What entities does a Version depend on to exist?
DISTREL	Distributions. A history of the Distributions comprised of a specific Version of a Manifest and its Location.
EVTLOG	Event Log is not currently used.
LICREL	Licences which have been secured for RMS/PrimeCode catalogs or remote Installs.
LOCKENT	Lock entities. i.e. Locked for Work, Edit. etc.

Table Name	Description
ORGENT	The organization of the Catalog as a group and details of each Group it contains as well as each Component in each Group. Records are created by any process which *links*, *compiles* or *adds* a Component.
PCRIREL	Any relationships between a UserChange Id (formerly Program Change Request) and Releases.
PCRVREL	Any relationships between a UserChange Id (formerly Program Change Request) and Versions of Components.
REFREL	References from Versions to other Components.
RELENT	Releases (collections of Components which can be referred to as one unit) that have been assembled.
REQREL	Requisites. Relationships between Releases and Requisites (Releases required to install prior to another Release).
RULEENT	Defined Compile Rules.
SCHEMVER	Versions of the Catalog. One record is created for each Catalog upgrade detailing its history.
SECENT	Security records for the Catalog. (Not ACP security) There may be two records for any given user. They will appear identical except for the Kind and Masks fields which will determine if the Security Record is *Allowing* Security or *Denying* it for the branch of the Catalog.
STATEENT	The established States and their Weights used to define stages in the SCM process being used.
STRREL	Relationships between the established States determining their order in the life cycle progression.
SVERREL	StateOfVersion contains one record per Version at a given State. Promoting a Component from one State to another causes modification of this file.
TEXTENT	Text is an entity containing all descriptive values in the system. Other entities can have links to text if descriptive values are needed. This file is analogous to the DDL DICTOTF file.

Table Name	Description
TEXTREF	
TEXTSUB	
TEXTVAL	
UCHGENT	User Change Identifiers or Program Change Requests.
VERSENT	Information on each Version within a Component.

In addition to these tables Custom Report Writer uses a number of auxiliary tables containing constant definitions used to replace numbers in the original tables with meaningful strings that are provided for the query views.

Table Name	Description
_CompilationRelationshipKinds	Traverse, NonTraverse, Document
_ComponentFlags	Bug Fix, Acquiring, DistribParams, Frozen, Configurable, IsChildDependant, ProgId
_ComponentKinds	Source, Document, Object, Group, Release, Data, Manifest, Subrlease, Variant, Location, SubSystem, Keyed, Blob
_ComponentModes	Normal, Release, External, System, Volatile, Reference. Most Components are of type Normal. System types cannot be changed by the user.
_CRWVERSION	Associates the SQL database with the appropriate CRW software version.
_DependancyKinds	Source, Data, Object, Processor, Release, Child
_DistributionKinds	Acquire, Distribute, Fallback, Install
_DistributionKinds2	Class Mask, Active Mask, Active, Partial, Superseded, Manual, Automatic

Table Name	Description
_Full Names of Components	Pulls information from the COMPENT table and produces a table of full Component names by traversing the Catalog tree back to the root. This table is used by many other queries that utilize full Component names.
_LicenseFlags	Inherit, NonInherit
_LockKinds	Edit, Work, Freeze, Read, Compile
_ReleaseEnvironmentKinds	
_ReleaseStates	Created, Content Frozen, Versions Frozen, Dropped
_RequisiteKinds	PreRequisite, CoRequisite, AntiRequisite, ContraRequisite, Replace Requisite
_SecurityKinds	Deny, Allow
_SecurityMasks	Admin, Promote, Demote, Manager, Secure, Thaw, Lock, Read, Freeze, Modify, Delete, Add
_TextRecordKinds	
_VersionFlags	Timestamp, Fingerprint, Disabled, Protected
_LAST UPDATE	The date stamp of the ASCII files used to create/update this database.

APPENDIX B

Queries (Views)

Predefined queries, created using the tables in the SQL database, are the foundation on which the preformatted reports are based. The following queries are de-normalized versions of the original tables, containing all their relevant values. It is suggested that you use these and other custom-made queries to create your own reports. Some sample reports are provided and are described in Appendix C.

Query Name	Description
Archive	De-normailzed version of table : ARCHENT. Produces a table listing the Archive Filename of each Component in the catalog along with its Logical Name.
AuxFile	De-normalized version of table : AUXENT. List of Auxiliary Attributes and their Data.
CompilationRules	De-normalized version of table : COMPREL. Produces a table listing Rules and the Source and Target Components linked by the Rule.
CompiledRules	De-normalized version of table : COMPDREL. Produces a table listing all Sources that have been compiled to Objects and the Rule used.
Component	De-normalized version of table : COMPENT. Produces a table with details for every Component in the Catalog.
Dependencies	De-normalized version of table : DEPREL. Produces a table identifying Dependencies and their Kind between Components.
Distributions	De-normalized version of table : DISTREL. Produces a table of all Locations to which Releases have been Installed or Distributed to.

Continued on next page...

Query Name	Description
EventLog	Not currently used.
LicenseFile	De-normalized version of table : LICREL. Produces a table of all Release-To-Location Licenses.
Locks	De-normalized version of table : LOCKENT. Produces a table of all Locked Components with all relevant details.
OrgTree	De-normalized version of table : ORGENT. For RMS internal use only.
PCRToRelease	De-normalized version of table : PCRIREL. Produces a table listing UserChanges included in Releases.
PCRToVersion	De-normalized version of table : PCRVREL. Produces a table listing UserChanges Linked to Component Versions.
References	De-normalized version of table : REFREL. Produces a table listing References and their Kinds between Components.
Releases	De-normalized version of table : RELENT. Produces a table with details about all Releases.
RequisiteFile	De-normalized version of table : REQREL. Produces a table listing Requisite Relationships and their Kinds between Releases.
Rules	De-normalized version of table : RELEENT. Produces a table with full listings of the text of all Rules.
SchemaVersion	De-normalized version of table : SCHEMVER. Produces a table with all the Versions of the RMS Catalog used to create the SQL database.

Continued on next page...

Query Name	Description
Security-Entry	De-normalized version of table : SECENT. Produces a table listing all Security Entries that exist in the RMS Catalog for every user.
StateOfVersion	De-normalized version of table : SVERREL. Produces a table listing all Component Versions and their State.
States	De-normalized version of table : STATEENT. Produces a table listing all State names available in the RMS Catalog.
StateTransitions	De-normalized version of table : STRREL. Produces a table listing pairs of States indicating valid transitions.
Text	De-normalized version of table : TEXTENT. Produces a table listing the full text of all descriptive values. The records in this table are used as part of the de-normalization process of the rest of the tables.
UserChange	De-normalized version of table : UCHGENT. Produces a table listing all User Changes Id's.
Versions	De-normalized version of table : VERSENT. Produces a table of different Versions of Components and details about them. The records in this table are used as part of the de-normalization process of the rest of the tables.

The following predefined queries are a result of some analysis of the original tables described in Appendix A. These too are the foundation on which the preformatted reports are based. It is suggested that you use these and other custom-made queries to create your own reports. Some sample reports are provided and are described in Appendix C.

Query Name	Description
_Components Accessed in the Last Month	Pulls information from the COMPENT table and produces a table of all Components accessed in the last month.
_Components Accessed in the Last Week	Pulls information from the COMPENT table and produces a table of all Components accessed in the last week.
_Components Created by Each User	Pulls information from the **COMPENT** table and produces a table of users and the number of Components created by each.
_Components Modified in the Last Month	Pulls information from the **COMPENT** table and produces a table of all Components modified in the last month.
_Components Modified in the Last Week	Pulls information from the **COMPENT** table and produces a table of all Components modified in the last week.
_Components without Descriptions by User	Intermediate query used by the following query: _Components Without Descriptions by User2
_Components without Descriptions by User2	Pulls information from the **COMPENT** table and produces a table of users and the number of Components created by each without providing a description for the Component. Total number of Components created by each user is also given, together with a percentage for Components without descriptions.

Continued on next page...

Query Name	Description
_Impact of Promoting Releases	Uses Visual Basic code [Function **RleaseImact(RID As String)**] to analyze the contents of a Release and calculate the Impact of any Release changes to its contents, and thus on the whole Catalog. **Note:** This may take a while.
_Physical Location of Components	Pulls information from the **COMPENT** table and produces a table of physical locations (\SYSTEM.$VOLUME.SUBVOLUME) and the number of Components that are stored there. This is a good way to figure out how far and wide a catalog extends.
_Release Contents	Uses Visual Basic code [Function **RleaseContents(RID As String)**] extract the contents of a Release and produces a table of all Releases and the Components they include.
_Type of Components Created by Date	Pulls information from the **COMPENT** table and produces a crosstab table with dates and counts of each type of Component created on that date.
_Type of Components Created by Month	Pulls information from the **COMPENT** table and produces a crosstab table with months and counts of each type of Component created on that month.

APPENDIX C

Reports

A number of sample reports are provided in the SQL database created by Custom Report Writer. These can be accessed from within MS Access.

Report Name	Description
CompilationRules	Each Rule known to the Catalog is listed in alphabetical order. Under each Rule name is listed the qualified name of the Sources and their associated Target Object which utilized this Rule.
Component Versions by State	States at which Components exist are listed. Beneath each State is a listing of every Component at that State and its Version. Component names are "qualified" to represent their location in the Catalog. Timestamp Version numbers are Objects.
Components by Language	Component names are organized by Language and then by Type.
Current Licenses by Location	The Location of the Release and the Release name are listed for every Licensed Release. Deny, Expiry Date and License flags are included.
Current Licenses by Release	Licensed Releases are sorted by Release and then Location.
Install Activity by Location	Locations are listed with each of the Releases which have been Distributed or Installed there.

Continued on next page...

Report Name	Description
Install Activity by Release	Releases are listed with each of the Locations to which they have been Installed or Distributed to.
Releases by Release State	All Releases are listed, sorted by the Release State. (i.e. Contents Frozen, Versions Frozen, etc.)
Rules	Rules for the Catalog are listed, sorted by the group at which they were placed. Details include the Rulename, State, Language and Steps.
Security-Entry	Catalog Security records for this Catalog are displayed, sorted by user.

A number of auxiliary report samples are included.

Report Name	Description
_Components Accessed in the Last Month	A listing of Components in this Catalog which have been accessed during the last month.
_Components Accessed in the Last Week	A listing of Components in this Catalog which have been accessed during the last week.
_Components Created by Each User - Graph	A 3-D pie chart representation of all the Components created, broken down by user.
_Components Created by Each User - Table	Component details of all Components created by each user.
_Components Modified in the Last Month	A table listing of all Components modified in the last month.
_Components Modified in the Last Week	A table listing of all Components modified in the last week.

Continued on next page...

_Components without Descriptions by User2	Information from the **COMPENT** table is used to produce a listing, by user showing Components created by each without a description. Total number of Components created by each user is also given, together with a percentage for Components without descriptions.
_Impact of Promoting Releases	Calculates the Impact of any changes to Release contents.
_Physical Location of Components	A listing of subvolumes and the number of Components which have their managed file at that location on disk.
_Release Contents	Releases are listed along with the Components which are included in them.
_Type of Components Created by Date	A crosstab, chronological listing by date of all Components that were created and the number of Components created on that date for each Component Type.
_Type of Components Created by Month	A crosstab, chronological listing by month of all Components that were created and the number of Components created in that month for each Component type.

APPENDIX D

Database Field Descriptions

ARCHENT

Table Description:

The archives of the RMS/PrimeCode Catalog contain copies of all Data Files, Objects which have been submitted and any compiled Objects.

Field Name	Type	Description
SYSTEM	String	The System portion of the archive name of the Component.
VOLUME	String	The Volume portion of the archive name of the Component.
SUBVOL	String	The SubVolume portion of the archive name of the Component.
FILENAME	String	The last portion of the archive name of the Component. [End of archive name]
VERSIONID	Number	The Component Version archived.
SYSTEM2	String	The managed filename of the component.
VOLUME2	String	
SUBVOL2	String	
FILENAME2	String	[End of managed filename]
VERSION	String / Date	The Version, or Version date, of the archived Component.

AUXENT

Table Description:

Auxiliary Attributes, many of which are user defined qualities of a Component.

Field Name	Type	Description
RECORDTYPE	Number	The type record UIDCOMPONENT, UIDRELEASE, UIDUSERCHANGE, UIDLOCATION or UIDLICENSE
IDENTID	Number	A unique identifier within RecordType.
LOGICALNAME	String	Attribute symbolic name.
SEQUENCE	Number	A sequence number to support multiple attributes with same logical name.
ATTRIBUTEDATA	String	The Value of the Attribute.

COMPDREL

Table Description:

Rules of compiled relationships - relationships defined between Source and Object Components which have been involved in a compile. This relationship is a permanent link between two Components. A Rule Id has been associated with this Rule superseding the Rule name.

Field Name	Type	Description
TARGETID	Number	A link to the target Version.
SOURCEID	Number	A link to the primary Source Version.
RULEID	Number	The Id of the Rule used.

COMPENT

Table Description:

Components.

Field Name	Type	Description
COMPONENTID	Number	A unique Component identifier.
PARENTID	Number	The Component Id of the primary parent of this Component. -1 is undefined. 0 is the root of the Catalog.
KIND	Number	The Components kind or type is used to determine some of the variations of actions upon them such as compilation. This field is the key in the table _Componentkinds which describes the kind Id number.
MODE	Number	A component type. Used to determine the nature of actions required during extract and release promotion.
DESCRIPTION	String	A brief description of the Component.
LANGUAGE	String	The language of the Component.
LOGICALNAME	String	A name which can be used in place of the Component Id and/or physical name.
SYSTEM	String	Physical System Name.
VOLUME	String	Disc/volume name.[$dat1].
SUBVOL	String	Subvolume name.
FILENAME	String	Last segment of physical file name [up to 8 characters].
SYSTEM2	String	Presentation Name.
VOLUME2	String	Presentation Name.
SUBVOL2	String	Presentation Name.

Continued....

COMPENT

Table Description:

Components.

Field Name	Type	Description
FILENAME2	String	Presentation Name.
GROUPID	Number	
DEFINENAME	String	The define name, a unique system wide alias, can be used in Source files extracted for compile instead of the Component id.
SYSTEMNUMBER	Number	Creator.
GROUPNUMBER	Number	Creator.
USERNUMBER	Number	Creator.
CREATIONDATE	Date	The date the Component was created.
MODIFICATIONDATE	Date	The date the Component was submitted.
ACCESSDATE	Date	The date the Component was last extracted or compiled.
FLAGS	Number	LOR of all special component flags.
DEFAULTEXTTRACTINCR	Number	Line number increments for extracted files.
DEFAULTTRIGGERCHAR	String	A character used to trigger embedded macros to expand. Range Null(0) to ~.
NUMCHILDREN	Number	Computed number of children. -1 is undefined.
STATUSTIME	Number	
STATUS	Text	
APPROVALREQUIRED	Number	

COMPREL

Table Description:

Rules of compile relationships - relationships defined between Source and Object Components which have not yet been used in a compile. A Rule name identifies the Rule and it can be edited or renamed at this point.

Field Name	Type	Description
TARTETID	Number	A link to the target Component.
SOURCEID	Number	A link to the primary Source Component.
RULENAME	String	A link to the Rule used to compile the target Version of the target Component. RuleName is resolved using the groupId of the SourceId.
KIND	Number	Specifies the type of the compile relationship (CompRelTraverse / CompRelNonTraverse / CompRelDocument) NOT USE.

DEPREL

Table Description:

Dependencies for each Version. What entities does a Version depend on to exist.

Field Name	Type	Description
TARGETID	Number	A link to the target Version.
SOURCEID	Number	A link to the Source Version.
KIND	Number	DepKindSource / DepKindObject / DepKindData / DepKindProcessor

DISTREL

Table Description:

Distributions. A history of the Distributions comprised of a specific Version of a manifest and its location.

Field Name	Type	Description
LOCATIONID	Number	A unique Component(location) identifier.
COMPONENTID	Number	The actual item (release or Component) which was distributed.
VERSIONID	Number	The actual Version of the release manifest which was distributed.
DISTRIBUTIONDATE	Date	A time when the Distribution occurred.
KIND	Number	DistTypeDistrib / DistTypeInstall / DistTypeAcquire / DistTypeFallback
SYSTEMNUMBER	Number	User responsible for Distribution.
GROUPNUMBER	Number	User responsible for Distribution.
USERNUMBER	Number	[end user]
REASON	Number	A link to the Reason for the Distribution.

EVTLOG

Table Description:

HP NonStop Event Log File. Not currently accessed by CRW.

Field Name	Type	Description
CLASS	Number	The class of the event.
EVENTNUMBER	Number	The number of the event.
EVENTDATE	Date/ Time	The date and time the event was generated.
TEXT	Text	The actual text of the event.

LICREL

Table Description:

Licences which have been secured for RMS/PrimeCode catalogs or remote installs.

Field Name	Type	Description
LICENSEID	Number	A unique license identifier.
RELEASEID	Number	The release licensed to the associated location.
LOCATIONID	Number	The location licensed for the associated release.
DENY	Number	yes/no
EXPIRYDATE	Date	The date the license expires.
LICENSEFLAGS	Number	LicenseInherit or LicenseNonInherit.

LOCKENT

Table Description:

Lock entities. i.e. Locked for work, edit. etc.

Field Name	Type	Description
COMPONENTID	Number	A Component identifier.
VERSION	String / Date	The Version, or Version date, of the Component which is locked.
LOCKKIND	Number	Edit/Work/Read/Compile/Freeze.
SYSTEMNUMBER	Number	The user who locked this Component.
GROUPNUMBER	Number	The user who locked this Component.
USERNUMBER	Number	The user who locked this Component.
LOCKDATE	Date	The date when this lock was created.
EXPIRYDATE	Date	The date and time after which the lock will be unrecognized. This primarily applies to read locks which can be cleaned up.
REASON	Number	A link to the reason for the lock.
USERCHANGEID	Number	The PCR being associated with the change.
EXTRACTFILE	String	The file to which the locked Component was extracted. This name either a local NonStop Kernel name or a network name. This is the last field and the null terminator must be included in the record. The records CAN be variable length.

ORGENT

Table Description:

The organization of the Catalog as a group and details of each group it contains as well as each Component in each group. Records are created by any process which *links*, *compiles* or *adds* a Component.

Note: StateId or VersionId will contain a value but never both as Versions do not have State ids or States, Version Ids.

Field Name	Type	Description
PARENTID	Number	A link to the parent Component of this entry.
LOGICALNAME	String	It is used to refer to a Component by State rather than by Version.
COMPONENTID	Number	A link to a specific Component.
STATEID	Number	A link to a specific State can be null or the STATEID.
VERSIONID	Number	A link to a specific Version can be null or the VERSIONID. NOTE: both State and Version are never set.

PCRIREL

Table Description:

Any relationships between a UserChange Id (formerly Program Change Request) and releases.

Field Name	Type	Description
RELEASEID	Number	A Release that included the associated UserChange.
USERCHANGEID	Number	A UserChange included in the associated Release.

PCRVREL

Table Description:

Any relationships between a UserChange Id (formerly Program Change Request) and Versions of Components.

Field Name	Type	Description
VERSIONID	Number	The VersionId that has the PCR linked to it.
USERCHANGEID	Number	The UserChangeId

REFREL

Table Description:

References from Versions to other Components.

Field Name	Type	Description
REFERENCEORID	Number	A link to the Source's Version.
REFERENCEEID	Number	A link to the Source Version.
KIND	Number	CompKindSource / CompKindObject / CompKindData.

RELENT

Table Description:

Releases (collections of Components which can be referred to as one unit) that have been assembled.

Field Name	Type	Description
RELEASEID	Number	A link to the Component Id acting as a place holder for the release. The logical name is stored in the Component place holder. The user who created this release is stored in the Component place holder. The creation date of this rel is stored in the comp pl.h.
STATEID	Number	State of the release. There should, eventually be one entry per release per State.
RELEASESTATE	Number	RelStateCreated / RelStateContentFrozen / RelStateVersionsFrozen / RelStateDropped.
CONTENTDATE	Date	If RelStateCreated then the date at which the contents of the release are scheduled to be frozen (optional). If NOT RelStateCreated then the date at which the contents of release were frozen.
VERSIONDATE	Date	If RelStateContentFrozen or RelStateCreated, the date at which the vers of all Components of the rel are scheduled to be frozen (optional). If RelStateVersionsFrozen or RelStateDropped, the date at which the Versions of all Components or of the rel were frozen.
PROMOTEDATE	Date	If RelStateVersionsFrozen or RelStateDropped then the date at which the release was last promoted. Otherwise NULL.
DROPDATE	Date	If RelStateDropped then the date at which the release was dropped. If NOT RelStateDropped then NULL.
INCLUDESOURCE	Number	Ship Source Components with the release
CHANGEID		
ENVIRONMENT		-1=GUARDIAN, 1=OSS

REQREL

Table Description:

Requisites. Relationships between Releases and Requisites (Releases required to install prior to another release).

Field Name	Type	Description
LEFTID	Number	A link to a Component id.
RIGHTID	Number	A link to a Component id.
REQUISITECLASS	Number	RelPreRequisite / RelCoRequisite / RelAnteRequisite / RelContraRequisite / RelReplaceRequisite.
MASK_S	Number	Evaluation type. Establishes the time at which the release is evaluated for requisites of all types. i.e. At Freeze, Acquire, Install, Fallback or All.
OPTIONAL	Number	evaluation only if release exists.

RULEENT

Table Description:

Defined Compile Rules.

Field Name	Type	Description
RULEID	Number	The unique Rule identifier.
GROUPID	Number	Group Id under which this Rule applies as a default.
RULENAME	String	A unique Rule name within the group.
STATEID	Number	Always null for Rule.
LANGUAGE	String	The language associated with a given Rule.
PROCESSORID_1	Number	The Component Id of the external compiler product used in the Rule can be null.
TEXTID_1	Number	A link to the actual command text.
PROCESSORID_...	Number	
TEXTID_...	Number	
PROCESSORID_64	Number	
TEXTID_64	Number	
ENVIRONMENT	Number	The target environment to which this Rule applies. This is a pointer to the environment which will be used during the compile. NOT USED.

SCHEMVER

Table Description:

Versions of the catalog. One record is created for each Catalog upgrade detailing its history.

Field Name	Type	Description
DICTVERSION	Number	In the form of “A00" The release of RMS. There may be multiple records in the file which represent the upgrade history of the RMS product for ”this" site.
VERSIONDATE	Date	The release date of DictVersion as specified by the RMS suppliers.
INSTALLATIONDATE	Date	The date on which DictVersion was installed.
REASON	Number	A link to information describing the nature of the release.

SECENT

Table Description:

Security records for the Catalog. (Not ACP security)

There may be 2 records for any given user. They will appear identical except for the Kind and Masks fields which will determine if the security record is *allowing* security or *denying* it for the branch of the Catalog.

Field Name	Type	Description
COMPONENTID	Number	Reference to the Component corresponding to this security entry.
STATEID	Number	The State Id for the Component which is secured. Nil may be specified for all States.
KIND	Number	SecDeny/SecAllow.
SYSTEMNUMBER	Number	The system on which the user must reside when using the specified Component. dRMS-VAL-ALLUSERS or dRMS-VAL-LOCALSYSTEM may be specified.
GROUPNUMBER	Number	The group number of the user having security access to a given Component. dRMS-VAL-ALLUSERS may be specified only if dRMS-VAL-ALLUSERS is specified in UserNumber.
USERNUMBER	Number	The user number of the user having security access to a given Component. dRMS-VAL-ALLUSERS may be specified.
MASK_S	Number	The type of access permitted.

STATEENT

Table Description:

The established States and their weights used to define stages in the SCM process being used.

Field Name	Type	Description
STATEID	Number	A unique State identifier.
STATENAME	String	A unique State name.
WEIGHT	Number	Importance of State.
STATECLASSID	Number	Class of which this State is a part.

STRREL

Table Description:

Relationships between the established States determining their order in the life cycle progression.

Field Name	Type	Description
STATEID	Number	A link to State identifier.
PRESTATEID	Number	A link to a State identifier which can precede StateId.

SVERREL

Table Description:

StateOfVersion contains one record per Version at a given State. Promoting a Component from one State to another causes modification of this file.

Field Name	Type	Description
CHANGEDATE	Date	The time at which the change occurred.
COMPONENTID	Number	Component id.
STATEID	Number	State id.
VERSIONID	Number	Version id.
SYSTEMNUMBER	Number	System number of user responsible for changing States.
GROUPNUMBER	Number	Group number of user responsible for changing States.
USERNUMBER	Number	User number of user responsible for changing States.
REASON	Number	A reason for promoting the Component.

TEXTENT

Table Description:

Text is an entity containing all descriptive values in the system. Other entities can have links to text if descriptive values are needed. This file is analogous to the DDL DICTOTF file.

Field Name	Type	Description
TEXTID	Number	A unique text identifier.
TEXTSEQ	Number	A sequence number for the text id. This field enables multiple commands to be executed for a given text id.
REFERENCECOUNT	Number	The number of instances of entities which reference the text item. This field is only relevant in the TextSeq = 0 record. When referenceCount = 0 it is safe to delete the textId. This could be done immediately or in a house keeping related function.
TEXT	String	The actual text string associated with the text id.

TEXREF

Table Description:

Field Name	Type	Description
VALUEPTR	Number	
HASH	Number	
REFERENCECOUNT	Number	

TEXTSUB

Table Description:

Relationships between the established States determining their order in the life cycle progression.

Field Name	Type	Description
TEXTRECKIND	Number	Either [TextRecKindOriginal] or [TextRecKindReference]
VERSIONID	Number	A unique version identifier
ORDINAL	Number	Ordinal line number where the substitution occurs
OFFSET	Number	Offset in the line where the substitution occurs
LENGTH	Number	Length of the text starting at offset to substitue for
VALUEPTR	Number	Pointer to the text

TEXVAL

Table Description:

Field Name	Type	Description
VALUEEN	Number	
VALUE	Text	

UCHGENT

Table Description:

User Change Identifiers or Program Change Requests.

Field Name	Type	Description
USERCHANGEID	Number	A unique PCR identifier.
CHANGENAME	String	A unique user defined name reference with a change.
DESCRIPTION	Number	A link to text associated with the user change identifier.
STATEID	Number	State of the user change.
BUGFIXSTATEDID	Number	State of the user change for bug fixes.
TESTCOMPONENTID	Number	NOT USED
AVAILABLETOGROUP	Number	The group Component Id in which this PCR can be referenced.

VERSENT

Table Description:

Information on each Version within a Component.

Field Name	Type	Description
VERSIONID	Number	A unique Version identifier.
COMPONENTID	Number	A link to the Component which references this record.
VERSION	String / Date	The Version, or Version date, of the Component which references this record.
CREATIONDATE	Date	The date when this Version was created.
SYSTEMNUMBER	Number	User who created the Version.
GROUPNUMBER	Number	User who created the Version.
USERNUMBER	Number	User who created the Version.
TRIGGERCHARACTER	String	The character which is used to trigger variable replacement when extracting the Component for compile of read access.
REASON	Number	A link to an explanation of why the Version was created.
VERSIONFLAG	Number	Details about field contents.
CHECKVALUE	String	A value used to check the integrity of the Version outside of the RMS “known” locations.
PRESENTATIONDATE	Date	The date corresponding to the “local” Version.
STATUSTIME		
STATUS		
APPROVALREQUIRED		