



Rocket Model 204 SirDBA

User's Guide

November 2014
DBA-0704-UG-02

Notices

Edition

Publication date: November 2014

Book number: DBA-0704-UG-02

Product version:

Copyright

© Rocket Software, Inc. or its affiliates 1999-2014. All Rights Reserved.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

This software and the associated documentation are proprietary and confidential to Rocket Software, Inc. or its affiliates, are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note: This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Corporate Information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA

Contacting Technical Support

If you have current support and maintenance agreements with Rocket Software and CCA, contact Rocket Software Technical support by email or by telephone:

Email: m204support@rocketsoftware.com

Telephone :

North America +1.800.755.4222

United Kingdom/Europe +44 (0) 20 8867 6153

Alternatively, you can access the Rocket Customer Portal and report a problem, download an update, or read answers to FAQs. You will be prompted to log in with the credentials supplied as part of your product maintenance agreement.

To log in to the Rocket Customer Portal, go to:

www.rocketsoftware.com/support

Contents

Proprietary Notices	ii
Contents	v
Chapter 1: Introduction to UL/SPF	1
UL/SPF packaging and installation requirements	2
Integrating UL/SPF with other subsystems	2
Related documentation	3
Chapter 2: SirDBA Setup	5
Chapter 3: Database Analysis Tools	7
Chapter 4: System Administration	9
Field mapping	9
Opening files	9
Specifying field mapping criteria	11
Displaying the mapped field.	13
Appendix A: Datetime Processing Considerations	15
Datetime Formats	15
Valid Datetimes	19
Datetime and format examples	19
Index	23

Introduction to UL/SPF

UL/SPF is a family of products implemented as User Language application subsystems and designed to function together in a *Model 204* Online. Each product in the family can be installed and run independently, or each can be installed and run as a component of the integrated *UL/SPF* (User Language / System Productivity Facilities) framework.

All *UL/SPF* products share a common "look and feel" that is modelled upon IBM's ISPF facilities. *UL/SPF* supplants the sometimes arcane command language of *Model 204*, providing a highly productive full screen interface to a variety of common *Model 204* functions. *UL/SPF* enhances the capabilities of *Model 204* and bypasses many of its restrictions, enabling the performance of routine tasks that were previously impossible or prohibitively time-consuming.

UL/SPF comprises the following products:

- SirDBA*** A system that analyzes *Model 204* databases to determine their logical structure, populating an internal catalog. *SirDBA* is distributed as a component of the *Sir2000 Database Analysis Tools*.
- SirFile*** A comprehensive facility both for monitoring the physical storage utilization of *Model 204* database files and for warning users of the need for file reorganizations. *SirFile* maintains historical information that allows it to predict when file sizing problems *will* occur, allowing a DBA to take preventative action before an application outage results.
- SirLib*** A system that provides change management and configuration control for *Model 204* User Language applications. Fully integrated within the programming environment, *SirLib* supports unique *Model 204* constructs such as file groups, while remaining nearly transparent to programmers.
- SirMon*** A comprehensive facility for monitoring the performance and availability of *Model 204* online systems. *SirMon* combines the real time monitoring of *Model 204* performance with intelligent full screen displays that facilitate System Manager duties.
- SirPro*** A collection of powerful and easy to use tools for programmers, database administrators, and application managers. *SirPro* provides programmers with powerful facilities for managing large libraries of User Language procedures, and it provides system managers with intuitive ISPF-like front ends to many *Model 204* system management commands.
- SirScan*** A high performance utility that allows users in a *Model 204* Online to browse the contents of its journal in real time. *SirScan* permits ordinary

users to view journal entries generated by their own online session, and it allows users in ADMIN SCLASSES to browse journal entries for any set of users. The data is displayed in a full-screen browser with powerful searching commands and filtering options.

In addition, a number of subsystems that are not linked into the *UL/SPF* menuing structure may be accessed via APSY-transfer from the *UL/SPF* applications. One such subsystem is **FACT**, a utility for browsing *SirFact* dumps.

There are also many sample web and client-server applications developed for the Janus product family that are distributed and installed along with the *UL/SPF* products. These include **JANCAT**, an application that builds normalized views of *Model 204* data for use by *Janus Specialty Data Store* applications, and **JANSSL**, a system for creating and managing SSL certificate requests.

1.1 UL/SPF packaging and installation requirements

All of the *UL/SPF* products are *Model 204* application subsystems written in User Language. *UL/SPF* is distributed as a set of *Model 204* files in a backup format produced by the *Model 204* DUMP command. All User Language based products are distributed in a single *Model 204* procedure file called **SIRIUS**.

UL/SPF makes extensive use of specialized User Language \$functions that enable the creation of User Language application systems that can support complex environments with minimal server size requirements. Prior to version 7.5 of *Model 204*, the \$functions were part of the *Sirius Mods*, the installation of which (*Sirius Mods Installation Guide*) was a prerequisite for any *UL/SPF* product.

The *UL/SPF* \$functions are included in the [http://m204wiki.rocketsoftware.com/index.php/List_of_\\$functions](http://m204wiki.rocketsoftware.com/index.php/List_of_$functions), and their use is controlled by a product authorization mechanism. Many functions require that a specific *UL/SPF* product be authorized. Some functions may only be invoked from an authorized procedure “signed” by Sirius Software or Rocket Software. The **SIRIUS** command examines the status of all Sirius products installed in a *Model 204* Online.

1.2 Integrating UL/SPF with other subsystems

UL/SPF and any of its constituent products can be easily integrated with other User Language subsystems. Whenever a *UL/SPF* component product is exiting, it first checks to see if the global variable **SIRIUS.COMM** exists and has a non-null value. If so, the *UL/SPF* product performs a subsystem transfer using the value in **SIRIUS.COMM** as the name of the target subsystem.

For example, the following code fragment lets you transfer into SIRMON. When SIRMON exits, control is transferred to the application subsystem MENUSYS, provided that NEXTPROC is the current subsystem's communication global variable:

```
%RC = $SETG('SIRIUS.COMM', 'MENUSYS')
%RC = $SETG('NEXTPROC', 'XFER')
%RC = $SETG('XFER', 'SIRMON')
STOP
```

In addition to the individual User Language subsystems that implement the *UL/SPF* products, an umbrella *UL/SPF* subsystem, **ULSPF**, provides a menu that contains entries for all *UL/SPF* components installed at a site.

The *UL/SPF* subsystems support fast path navigation. For example, a *SirMon* user can transfer into *SirPro* Option 1 by typing:

```
=M.1.1
```

1.3 Related documentation

The following documentation is available from <http://docs.rocketsoftware.com> (M204 folder) or from <http://m204wiki.rocketsoftware.com>:

- ***SirDBA User's Guide***
- ***SirFile User's Guide***
- ***SirLib User's Guide***
- ***SirMon User's Guide***
- ***SirPro User's Guide***
- ***SirScan User's Guide***
- ***Sirius Mods Installation Guide***
- ***Model 204 System Manager's Guide***

The following documentation is available from the Model 204 documentation wiki (<http://m204wiki.rocketsoftware.com>):

- UL/SPF installation
(http://m204wiki.rocketsoftware.com/index.php/UL/SPF_installation_guide)
- \$functions
(http://m204wiki.rocketsoftware.com/index.php/M204wiki_main_page#.24Functions)
- Model 204 commands
(http://m204wiki.rocketsoftware.com/index.php/List_of_Model_204_commands)
- Model 204 files
(http://m204wiki.rocketsoftware.com/index.php/Category:File_architecture_and_management)
- UL/SPF product messages
(http://m204wiki.rocketsoftware.com/index.php/Category:Sirius_Mods_messages)
- User Language/SOUL (<http://m204wiki.rocketsoftware.com/index.php/Category:SOUL>)

CHAPTER 2 ***SirDBA Setup***

The installation process for *SirDBA* is described in M204wiki at http://m204wiki.rocketsoftware.com/index.php/UL/SPF_installation_guide.

In addition, because *SirDBA* is an adjunct product to *SirPro*, the *SirPro* application must be installed in the same *Model 204* region(s).

CHAPTER 3 ***Database Analysis Tools***

SirDBA maps the fields in *Model 204* files and groups onto standard data types, and it stores the mappings in a database file. When a field is selected from the *SirDBA* field display, you are launched into *SirPro*'s string search facility to hunt for procedures that reference the specified field name.

SirDBA is designed to assist in database conversions and Year 2000 projects, and for conversions involving a large volume of changes to User Language code, when the changes are centered around *Model 204* field names.

For Year 2000 conversion, *SirDBA* goes to extra lengths to determine when a field contains date data. When it does, *SirDBA* stores a datatype of DATETIME and a format specification of the datetime field.

Because *SirDBA* is an adjunct product to *SirPro*, programmers should familiarize themselves with the feature list in the ***SirPro User's Guide*** Using *SirDBA* and *SirPro*, a User Language development team has the tools for targeted large scale procedure updates centered around references to *Model 204* fields.

4.1 Field mapping

If *Model 204* supported strong data typing at the field level, *SirDBA* could simply allow programmers to work from a list of field characteristics. Instead, most *Model 204* fields are nominally formatted as character strings, but they may contain anything the application places in them.

SirDBA's mapping facility processes the actual file contents to determine the type of data each field contains. The mapping facility requires the user to select the data file or group, and to specify a few characteristics of the mapping job. *SirDBA* then analyzes a sample of the file's records to determine what's actually in the fields, and it stores an entry for the file/field combination in its internal database. These database entries are used by the *SirDBA* screens to let you select fields to work on.

To run *SirDBA* and map the fields in a *Model 204* data file, you simply:

1. Open the target database file.
2. Specify the number of records to scan and any specific date formats known to be in the file.

The *SirDBA* mapping facility will typically take a few minutes to run.

3. Select fields from the *SirDBA* display lists (which transfers them into a *SirPro* edit session for the procedures that contain references to the selected field).

4.2 Opening files

When you first enter *SirDBA*, no files will have been mapped. Rather than display an empty selection list, the first user must specify the first file to be mapped.

```
----- Open a file or group -----98-04-22 16:46:59--

Context          ==> FILE
File/Group name  ==> SIRLIBD
Password         ==>
Current Privileges ==>
Required Privileges ==> 0041

----- DBA.009: Unable to open FILE SIRLIBD.
1/Help          3/Quit
```

Opening a Model 204 File

This screen is also presented when a file is added to the *SirDBA* database and *SirDBA* can't open the file with high enough default privileges.

- | | |
|----------------------------|--|
| Context | Specify FILE or GROUP for the target <i>Model 204</i> entity. |
| File/Group name | Name of the <i>Model 204</i> data file or data group whose fields are to be scanned. |
| Password | If a password is not required, <i>SirDBA</i> will usually open the file without sending the user to this screen. |
| Current Privileges | Displays the hexadecimal privilege switches, if the target file is already open by the user without sufficient privileges for <i>SirDBA</i> to complete the mapping. |
| Required privileges | Hexadecimal privilege level required for <i>SirDBA</i> to perform field mappings. Currently hardcoded to <code>0041</code> . |

PF keys:

- PF1** Help.
- PF3** Quit / Return to previous screen.

4.3 Specifying field mapping criteria

Field mapping requires the user to specify the number of records to scan, a "fudge" factor for estimating field lengths and some date formats. The screen for specifying these settings is presented once the target file is open.

```

----- SIRDBA Model 204 Field Analyzer -----98-04-22 15:58:36

Map tables for FILE METADATA

Records to scan    ==> 500
Char fudge factor ==> 2

                Date/Time formats to scan for in file
-----
==> YYYYMMDD                ==> DD MON YYYY
==> YYYYDDD                 ==> YYYYMMDDYYMM
==> YYMMDD                  ==> YYIMMIDDIHHIMMISS
==> YYDDD                   ==> YYYYMMDDHHMI
==> YYYY MONTH DAY         ==> MMIDDIYY
==> YY MONTH DAY           ==> DDIMMIYY
-----

1/Help                3/Quit                5/Restore Defaults
                                     12/Submit

```

Specifying Mapping Criteria

Input and display options on this screen are:

Map tables for FILE/GROUP

Displays the name of the current target data file.

Records to scan

The number of records *SirDBA* will analyze to determine the data formats and lengths for each field. Analyzed records are selected evenly throughout the file. So, for instance, if the target file contains 60,000 records, and the user specifies that 1000 should be analyzed, every 60th record is examined (using *Model 204* internal record numbers).

Enough records should be analyzed to guarantee that each field in the file is represented in the *SirDBA* database. For instance, if a file contains a million records, and some fields exist only on a record type that occurs 1000 times in the file, you'll need to sample more than every thousandth record in order to capture the contents of that record type.

The number of records may be set from 1 to 99999. There is no penalty for oversampling, except that it will take *SirDBA* longer to complete the analysis.

CHAR fudge factor

SIRDBA calculates the length of each column when building tables by keeping track of the average and longest lengths for every observed occurrence of the field. It also calculates the standard deviation for the set of observations.

The default column length is set to the observed mean plus two standard deviation units, which should be an adequate length for 99% to 100% of occurrences of the field.

Note: The algorithm using +2 standard deviations works even when fields are a fixed length, as there will be no variance from the mean, and the standard deviation will be 0. The "fudge factor" of 2 standard deviations is almost always the preferred value.

Date/time formats

While scanning records to build the field mapping, *SirDBA* will map a field to a datetime column if the contents of the field map consistently to a supported datetime format. *SirDBA* defaults to search for the following format fields:

```
YYYYMMDD
YYYYDDD
YYMMDD
YYDDD
YYYY MONTH DAY
YY MONTH DAY
DD MON YYYY
MON DD YYYY
YYIMMIDDIHHIMMISS
YYYYMMDDHHMMSS
```

You can override the defaults and specify as many as 12 date formats for *SirDBA* to look for. Date/Time formats must be 31 characters or shorter; see [“Datetime Formats” on page 15](#) for a description of the possible formats.

PF keys:

- PF1** Help.
- PF3** Quit / Return to previous screen.
- PF9** Quit / Reset the system defaults (the default number of records to scan, CHAR fudge factor = 2, and the 10 default date formats).
- PF12** Submit the request to map the file.

4.4 Displaying the mapped field.

This is the main *SirDBA* screen showing files and fields that have been mapped onto data types. This is also the main screen that displays when *SirDBA* is entered and any file has already been mapped.

From here, if you place an "S" in the selection field in front of any line, you transfer into a *SirPro* edit session for procedures that contain references to that field.

To add a new file to *SirDBA*, simply type "ADD filename" on the command line at the top of the screen, or press **PF 12**.

```

----- SIRDBA Field Mappings by File -----98-04-22 16:48:15--
==>                                     File: 1 of 83
  File/Grp Field Name                    Type      Format
_ ALANPROC A B                          CHAR(70)
_ ALANPROC RECORD TYPE                  CHAR(1)
_ ALANPROC SELECT                       CHAR(22)
_ ALANPROC TESTDATE1                   DATETIME YYYYMMDD
_ ALANPROC TESTDATE2                   DATETIME MMIDDIYY
_ ALANPROC X                            CHAR(14)
_ ALANPROC Y                            CHAR(11)
_ ALANPROC Z                            DATETIME YYYYMMDDHHMISS
_ METADATA ADMIN PROC                  CHAR(40)
_ METADATA ADMIN TITLE                 CHAR(70)
_ METADATA ALIAS                       CHAR(6)
_ METADATA AVERAGE LENGTH             INT
_ METADATA AVERAGE OCCURS             INT
_ METADATA CODED                      CHAR(1)
_ METADATA COMMAND                    CHAR(6)
_ METADATA COMMAND STATE               CHAR(1)
_ METADATA CREATE DATE                 CHAR(9)
-----*** Top of data ***
1/Help      2/Sort-Field 3/Quit      4/Sort-File  6/Sort-Type
7/Up        8/Down      9/Repeat   10/Refresh   12/Map File

```

Displaying mapped fields

Input/Display options on this screen are:

File/Group Name: Name of the file or group in which *SirDBA* found the field.

Field Name: Name of the field. Although the entire field name is stored in the *SirDBA* database, only the leftmost 25 characters of the field name are displayed.

Type: Type of data *SirDBA* found in the field when scanning the file or group. Data types are:

CHAR Character data. The data length is shown in parentheses.

DATETIME Date/Time data.

FLOAT Numeric data with a decimal.

INT Integer numeric data.

Since *SirDBA* scans only the number of records you tell it to, it can make bad guesses about data types if you scan too few records and certain field values are clustered near the end of the file.

Format: Formats for DATETIME fields are displayed.

Format is only shown when the data type is DATETIME. See [“Datetime Formats” on page 15](#) for a description of the possible formats.

Prefix Commands:

S Selects the field and transfers the user to SIRPRO to begin editing procedures that use it.

The following commands may be entered on the Command line:

ADD <context> <file/group>

ADD is used to invoke the AUTOBUILD feature to add field mappings for a designated file or group. Groups must be permanent groups.

QUIT, STOP, END or LOG:

Perform the same function as **PF3**, which exits *SirDBA*

PF keys:

- PF1** Help.
- PF2** Sort the displayed list by Fieldname.
- PF3** Quit.
- PF4** Sort the displayed list by File/Groupname, then Fieldname.
- PF6** Sort the displayed list by Field Type.
- PF7** Scroll the display up.
- PF8** Scroll the display down.
- PF9** Repeat the last command
- PF10** Refresh the list.
- PF12** Map the fields in a file to *SirDBA* field types.

APPENDIX A ***Datetime Processing Considerations***

This chapter presents date processing issues, including usage of *SirDBA* past the year 1999, an explanation of its processing of dates, and any rules and restrictions you must follow to achieve correct results using date values with *SirDBA*.

SirDBA uses dates in the following ways:

- To examine the CPU clock (as returned by the STCK hardware instruction) to determine the current date, in case *SirDBA* is under a rental or trial agreement
- As values to check for field type mapping
- To display the current date, as returned by the TIME SVC, modified by the SYSDATE parameter or the the *Sir2000 User Language Tools* APPDATE clock, as page headers in various end-user displays

To correctly use *SirDBA* past the year 1999, version 5.0 of the *Sirius Mods* and version 5.0 of *UL/SPF*, or later, are required. For headers on pages or rows that occur on printed pages or displayed screens, Sirius Software products generally use a full four digit year format, although they may display dates with two digit years in circumstances where the proper century can be inferred from the context.

Above and beyond the post-1999 requirements specific to *SirDBA*, you must examine all uses of date values in your applications to ensure that each of your applications produces correct results. Furthermore, both the operating system and *Model 204* must correctly process and transmit dates beyond 1999 in order for *SirDBA* to operate properly.

The rest of this chapter contains a discussion of datetime formats and valid datetime strings. and processing of two-digit year values. It also contains example datetime formats and corresponding example datetime strings.

A.1 Datetime Formats

The representation of a date is determined by a *datetime format*. This value is a character string, composed of the concatenation of tokens (for example, "YYYY" for a four-digit year, and "MI" for minutes) and separator characters (for example, "/" in "MM/DD/YY" for two-digit month, day, and year separated by slashes).

These *datetime format* strings are used in many products in addition to *SirDBA*. The products using datetime format strings are:

- *Fast/Unload*
- *Janus Open Client*
- *Janus Open Server*
- *Janus Specialty Data Store*
- *Janus Web Server*
- *SirDBA*
- *Sirius Functions*
- *Sir2000 Field Migration Facility*
- *Sir2000 User Language Tools*

The rules for these *datetime format* strings are consistent throughout all these products, though certain uses of these strings might impose extra restrictions. For example, a leading blank may match an HH, DD, or MM token in *SirDBA*, but it may not in some cases in other Sirius products.

There are certain rules applied to determine if a format is valid. The basic rules are:

1. If a format string contains a numeric datetime token (i.e. "ND", "NM", or "NS"), then the format string must consist of only one token. Numeric datetime tokens are only supported in format strings for the *Sir2000 Field Migration Facility*.
2. You must specify at least one time, weekday, or date token.
3. Except for "weekday", you can't specify redundant information. More specifically this means
 - Except for "l", no token can be specified twice.
 - At most one year format (contains Y) can be specified.
 - At most one month format (contains MON, Mon, or MM) can be specified.
 - At most one day format (DD or Day) can be specified.
 - At most one weekday format (WKD, Wkd, WKDAY, or Wkday) can be specified.
 - If AM is specified, then PM can not be specified.
 - At most one fractions-of-a-second format (contains X) can be specified.
 - If DDD is specified, then neither a day nor month format can be.
4. If ZYY is specified in a format string, no other token that denotes a variable-length value may be used.
5. If a format string contains other tokens that denote variable length values, then an * token may only appear as the last character of the format string.

6. The DAY token may not be immediately followed by another token whose value may be numeric, regardless of whether the following token represents a variable length value. Thus, DAY may not be followed by *, I, YY, YYYY, CYY, MM, HH, MI, SS, X, XX, or XXX; DAY may not be followed by a decimal digit separator, and DAY may not be followed by a quote followed by a decimal digit.
7. The maximum length of a format string in *SirDBA* is 31 characters; in most other Sirius products it is 100 characters.

Note: A common mistake is to use "MM" for minutes; it should be "MI".

The valid tokens in a date format are shown in the following list. All of the tokens which match alphabetic strings (for example, "MON") match any case string (for example, "jan" or "JAN" or "Jan").

NM	numeric datetime value containing the number of milliseconds (1/1000 of a second) since January 1, 1900 at 12:00 AM. (This token is allowed only in the <i>Sir2000 Field Migration Facility</i> .)
NS	numeric datetime value containing the number seconds since January 1, 1900 at 12:00 AM. (This token is allowed only in the <i>Sir2000 Field Migration Facility</i> .)
ND	numeric date value containing the number of days since January 1, 1900. (This token is allowed only in the <i>Sir2000 Field Migration Facility</i> .)
*	Ignore entire variable-length substring matching pattern, if any. See “Datetime and format examples” on page 19 .
I	Ignore corresponding input character. See “Datetime and format examples” on page 19 .
"	Following character is "quoted", that is, it acts as a separator character. See “Datetime and format examples” on page 19 .
YYYY	Four-digit year
YY	Two-digit year
CYY	Year minus 1900 (three digits, including any leading zero). See “Datetime and format examples” on page 19 .
ZYY	Year minus 1900, two-digit or three-digit year number (variable length data). See “Datetime and format examples” on page 19 .
MONTH	Full-month name (uppercase variable length). For <i>SirDBA</i> , this is the same as Month.
Month	Full-month name (mixed-case variable length). For <i>SirDBA</i> , this is the same as MONTH.
MON	Three-character month abbreviation (uppercase). For <i>SirDBA</i> , this is the same as Mon.
Mon	Three-character month abbreviation (mixed case). For <i>SirDBA</i> , this is the same as MON.
MM	Two-digit month number. For <i>SirDBA</i> , this is the same as BM (leading blank is allowed). See “Datetime and format examples” on page 19 .
BM	Two-character month number; for <i>SirDBA</i> , this is the same as MM. See “Datetime and format examples” on page 19 .
DDD	Three-digit Julian day number

DD	Two-digit day number. For <i>SirDBA</i> , this is the same as BD (leading blank is allowed). See “Datetime and format examples” on page 19 .
BD	Two-character day number; for <i>SirDBA</i> , this is the same as DD. See “Datetime and format examples” on page 19 .
DAY	One-digit or two-digit day number (variable length data). See “Datetime and format examples” on page 19 .
WKDAY	Full day of week name (uppercase variable length). For <i>SirDBA</i> , this is the same as Wkday.
Wkday	Full day of week name (mixed case variable length). For <i>SirDBA</i> , this is the same as WKDAY.
WKD	Three-character day of week abbreviation (uppercase). For <i>SirDBA</i> , this is the same as Wkd.
Wkd	Three-character day of week abbreviation (mixed case). For <i>SirDBA</i> , this is the same as WKD.
HH	Two-digit hour number. For <i>SirDBA</i> , this is the same as BH (leading blank is allowed). See “Datetime and format examples” on page 19 .
BH	Two-character hour number; for <i>SirDBA</i> , this is the same as HH. See “Datetime and format examples” on page 19 .
MI	Two-digit minute number
SS	Two-digit second number
X	Tenths of a second
XX	Hundredths of a second
XXX	Thousandths of a second (milliseconds)
AM	AM/PM indicator
PM	AM/PM indicator

The valid separators in a date format are:

- blank (" ")
- apostrophe ("'")
- slash ("/")
- colon (":")
- hyphen ("-")
- back slash ("\")
- period (".")
- comma (",")
- underscore ("_")
- left parenthesis ("(")
- right parenthesis (")")
- plus ("+")
- vertical bar ("|")
- equals ("=")
- ampersand ("&")
- at sign ("@")
- sharp ("#")
- the decimal digits ("0" - "9").

In addition, any character may be a separator character if preceded by the quoting character (").

See [“Datetime and format examples”](#) for examples which include use of various separator characters.

A.2 Valid Datetimes

For a datetime string to be valid it must meet the following criteria:

- Its length must be less than 128 characters.
- It must be compatible with its corresponding format string.
- It must represent a valid date and/or time. For example, at most 23:59:59.999 for a time, 01-12 for a month, 01-31 or less (depending on the month) for a day, February 29 is only valid in leap years (only centuries divisible by 4 are leap years: 2000 is but neither 1800, 1900, nor 2100 are). Note: weekdays are not checked for consistency against the date; for example, both Saturday, 02/15/97 and Friday, 02/15/97 are valid.
- It must be within the date range allowed for the corresponding format. A datetime string used with a CYY or ZYY format can only represent dates from 1900 to 2899, inclusive. The valid range of dates for all other formats is from 1 January 1753 thru 31 December 9999.

A.3 Datetime and format examples

There is an extensive set of format tokens, as shown in [“Datetime Formats” on page 15](#). These tokens and the various separator characters can be combined in almost limitless possibility, giving rise to an extremely large set of datetime formats. This section provides examples of some common datetime formats, and also tries to explain the use of some of the format tokens which might not be obvious. Each example format is explained and also presented with some matching datetimes; again, bear in mind that these tokens can be combined in very many ways and only a very few are shown here.

YYMMDD This is the common 6-digit date format which supports sort order if all dates are within a single century. The value 960229 matches this format.

YYYYMMDD

This is the common 8-digit date format which supports sort order with dates in 2 centuries. The value 19921212 matches this format.

MM/DD/YY

This is the U.S. 6-digit date format for display. The value 12/14/94 matches this format.

Notes:

- The leading zero corresponding to an MM token may be given as a blank, thus allowing “ 7/15/98”.
- The BM token can be used instead of MM.

DD.MM.YY

This is a European 6-digit date format for display. The value 14.12.94 matches this format.

Notes:

- The leading zero corresponding to a DD token may be given as a blank, thus allowing “ 7.04.89”.
- The BD token can be used instead of DD.

Wkday, DAY Month YYYY "A" T HH:MI

This is a format which would be quite unlikely in a file, but just to round out the discussion of some date format tokens, the value “Friday, 7 February 1998 AT 21:33” matches this format.

Notes:

- If a format contains AM or PM, then the time (HH:MI) must be between 00:01 and 12:00 and must be accompanied by either AM or PM.
- The day number (string matching DAY) may have a leading zero.
- The leading zero corresponding to an HH token may be given as a blank, thus allowing

Friday, 31 February 1998 AT 8:33

- The BH token can be used instead of HH.

YYIIII

This is a format which could be used for a field which contains a 2-digit year prefixing other information, such as a sequence number. The value 92A123 matches this format.

YY*

This is a format which could be used for a field which contains a 2-digit year prefixing other information, such as a sequence number, when the other information is variable length. The values 92 and 92ABC123 match this format.

Note:

- At most one occurrence of the * token may appear in a datetime format.

CYYDDD This is a compact 6-digit date format with explicit century information, from 1900 through and including 2899. The value 097031 (representing 31 January 1997) matches this format.

ZYYMMDD

This is a compact 6- or 7-digit date format with explicit century information, from 1900 through and including 2899, that can often be used with “old” YYMMDD date values in the 1900's. The values 970501 (representing 1 May 1997) and 1000501 (representing 1 May 2000) match this format.

Notes:

- The century and year number (string matching ZYY) may have a leading zero.

YY0000 Decimal digits can be used as separator characters. The value 980000 matches this format.

Notes:

- Numeric separators, unlike alphabetic separators, do not need to be preceded by a quote character (").

Index

A

Adding files to *SirDBA* ... 13

D

Date processing ... 15

F

Field display ... 12

Files

 opening ... 9

 specifying field mapping criteria ... 10

S

SirDBA Setup ... 5

System Administration
 mapping fields ... 7

U

UL/SPF

 constituent products ... 1

 integrating with other subsystems ... 2

 introduction to ... 1

