

SYSRDC Utility

The SYSRDC utility is used to record monitoring and accounting data on the internal process flow within a Natural application. This data can be used for evaluating Natural session activities in external or Natural programs. The data is collected at events performed within Natural.

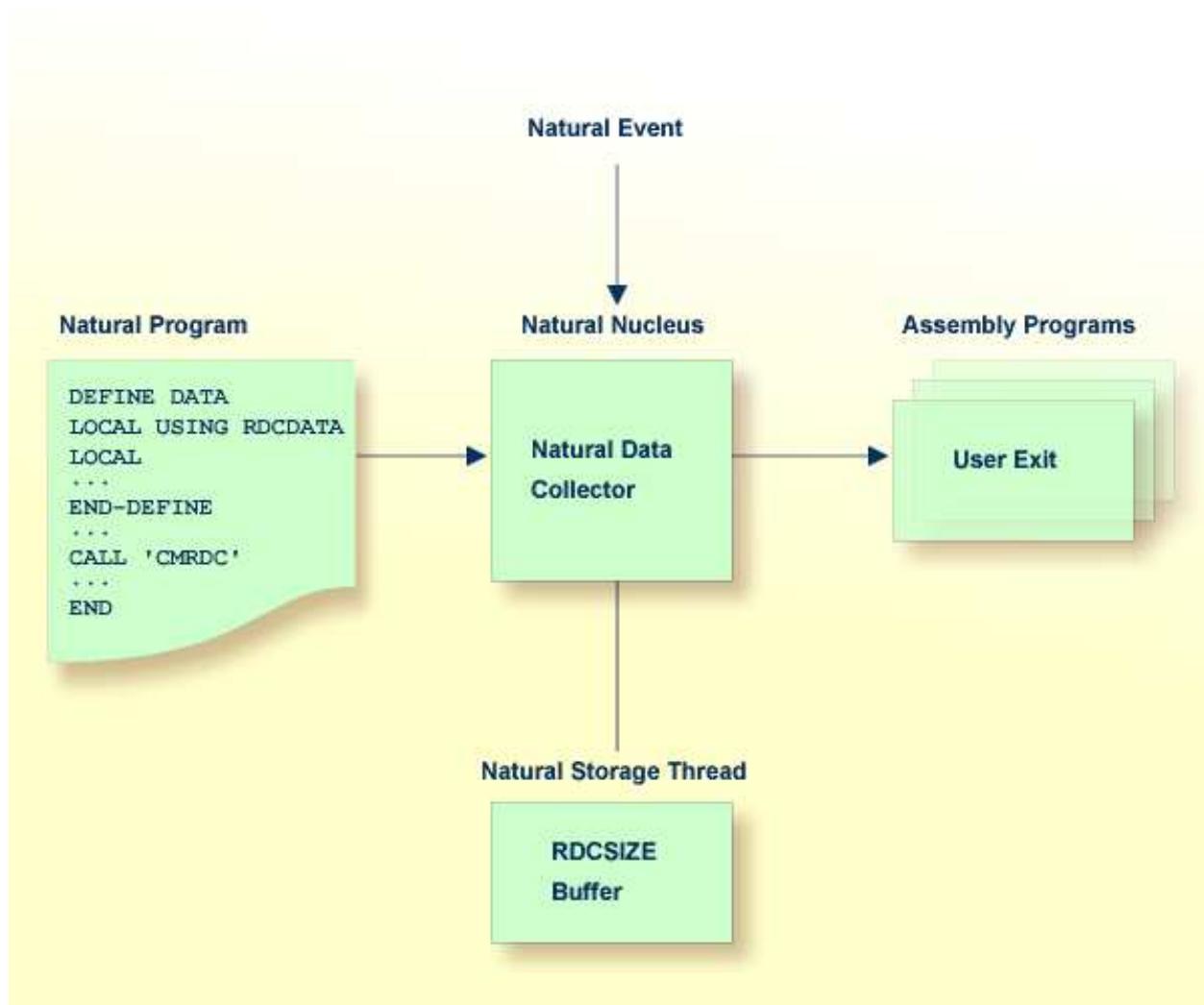
The *SYSRDC Utility* documentation covers the following topics:

- Functional Components of SYSRDC
 - Data-Collecting Events
 - Data Collected
 - Activating the Natural Data Collector
 - Trace Recording
 - User Exits for External Monitoring/Accounting
 - Calling the CMRDC Interface
-

Functional Components of SYSRDC

The components SYSRDC utility provides to collect data and supply it for further processing are listed and illustrated in the following section:

- **Natural Data Collector as part of the Natural nucleus:**
Collects data and controls data recording in the RDCSIZE buffer.
- **User exits:**
Passes data to external monitoring and accounting programs.
- **CMRDC interface:**
Evaluates data in a Natural program for the current Natural session.



Data-Collecting Events

Events specify activities within Natural. The table below lists the events at which data is collected within Natural and the types of event available.

Each type of event is assigned a one- or two-letter event code where the first letter represents the type of event and any second letter the subtype.

For example, in the event code **PL**, **P** represents the event type "program" and **L** the subtype "load".

Event	Event Code
At session initialization.	SI
At session termination.	ST
At program load.	PL
At program start.	PS
At program termination.	PT
Before a database call.	DB
After a database call.	DA
Before a terminal I/O.	IB
After a terminal I/O.	IA
Before a call of a non-Natural program.	CB
After a call of a non-Natural program.	CA
At a runtime error.	E
At an internal trace call. The information internal traces provide are only intended for debugging purposes by Software AG personnel. See also "To activate an internal trace call event" below.	N
At a Natural statement.	NS
At outbound RPC message	RO
At inbound RPC message	RI
At start of RPC request execution	RS
At a user-defined event	U

To activate an internal trace call event

1. Specify the Natural profile parameter ITRACE=ON.
2. Define the Natural components that are to issue internal trace calls:
 - By using the NTTRACE macro of the Natural parameter module.

Or, by setting the dynamic Natural profile parameter TRACE.

The NTTRACE macro and the parameters ITRACE and TRACE are described in the *Parameter Reference* documentation.

Data Collected

The data the Natural Data Collector collects at events are described in the following section.

For the layout of the data, see the Natural source dataset NAMRDC, or the local data area RDCDATA that is provided in the Natural system library SYSRDC.

The data collected can be split into two categories: general and event-specific data:

- General Data
- Event-Specific Data

General Data

The following general data is collected at every event:

- Software AG product name
- Product version
- Operating system
- TP monitor
- Run modes, such as addressing or residence mode of the Natural nucleus and the buffer pool.
- TP user or batch job name
- TP terminal ID
- Current Natural user ID
- Current Natural Security user group ID
- Current Natural library
- Current Natural program,
- Current program level,
- Line number of program statement currently executed.

Event-Specific Data

The following data is only collected at the following events:

Event	Data Elements
Session initialization	none
Session termination	Termination return code. Natural termination message code NAT99nn. Name of back-end program.
Program load	Name of program to be loaded. Name of load library.

Event	Data Elements
Program start/termination	Program type. Program name. Program library name. Database ID of program library.
Database call	Database type. Command code. Command ID. Database ID. Response code. Pointer that lists parameter addresses. (Only useful for user exits.)
Terminal I/O	Number of bytes sent. Number of bytes read. Total session storage allocated. Compressed session storage length.
Call of non-Natural program	Name of program called. Calling mode, such as dynamic or static mode. Program link location. Parameter type. Parameter address. Program entry address. Response code.
Runtime error	Natural system error message code. External abend code Name of error handling program as specified with the Natural profile parameter ETA (see the relevant section in the <i>Parameter Reference</i> documentation).
Internal trace	Up to 250 bytes of information on Natural nucleus components. The information provided is only intended for debugging purposes by Software AG personnel.

Event	Data Elements
RPC request	RPC-specific information: Environment: C = client, S = server. RDC subtype S: subprogram name, Adabas user ID (ETID), conversation status, logon indicator (Y = logon performed), impersonation indicator of RPC request (Y = impersonation performed). RDC subtypes O and I: transport protocol, RPC function, type of client user ID, length of message, return code of transport layer, external conversation ID, client user ID, server node (outbound message of client only), server name (outbound message of client only).
User-defined event	Up to 250 bytes of user-defined information.

Activating the Natural Data Collector

The Natural Data Collector is activated and controlled by the Natural profile parameter RDCSIZE (see also the relevant section in the *Parameter Reference* documentation).

This parameter determines the size (in KB) of the RDCSIZE buffer for the Natural Data Collector and controls trace recording as described in the relevant section.

By default, the parameter is set to RDCSIZE=0, which means that the Natural Data Collector is deactivated.

To activate the Natural Data Collector

- Set RDCSIZE=2.

The Natural Data Collector is activated and the data collected is supplied to the user exits. No data is recorded in the RDCSIZE buffer.

To record data in the RDCSIZE buffer, see Trace Recording below.

Trace Recording

The event data collected is always supplied to user exits to be used by external monitoring/accounting programs as described in the relevant section. Independently of the user exists, you can record the event data of the current Natural session in the RDCSIZE buffer. This can be useful for testing purposes. In this section, the recording of data in the RDCSIZE buffer is referred to as trace recording.

Trace recording starts when initializing a session and ends when terminating a session. The data recorded in the RDCSIZE buffer can be accessed by any Natural program within the same session.

The RDCSIZE buffer is filled in wrap-around mode; that is, the oldest record is overwritten when the buffer becomes full. At the end of the session, the contents of the buffer are deleted.

Trace recording is activated with the profile parameter RDCSIZE (see also the relevant section in the *Parameter Reference* documentation). This parameter also determines the size of the RDCSIZE buffer.

To activate and use trace recording

1. Set the Natural profile parameter RDCSIZE to a value greater than 2.

Use the following formula to calculate the number of records that fit into the RDCSIZE buffer:

$$11 * (RDCSIZE - 2) = \text{number of records}$$

All event data collected for the current Natural session is recorded in the RDCSIZE buffer.

2. Retrieve the contents of the RDCSIZE buffer by using the CMRDC interface as described in *Calling the CMRDC Interface*.
3. If desired, use the CMRDC interface to stop or restart and trace recording in the RDCSIZE buffer, specify events for which data is to be recorded or execute additional functions described in *Calling the CMRDC Interface*.

User Exits for External Monitoring/Accounting

The event data can be passed to external monitoring and accounting programs for evaluation of activities in Natural sessions. This is accomplished with the user exits and the examples of user-exit programs provided. User-exit programs are written in assembly language.

The Natural Data Collector supports any number of user exits. A user exit can be defined by the Natural profile parameter RDCEXIT (see the relevant section in the *Parameter Reference* documentation). An external monitoring/accounting program can be attached to each user exit.

Three exit names are predefined: RDCEX1, RDCEX2 and RDCEX3. If you use one of these names as entry point for your exit (which is linked to the Natural nucleus), the Natural profile parameter RDCEXIT is not required.

At every event listed under *Data-Collecting Events*, the user exits take over control by using the standard linkage call conventions described in the table below:

Register	Contents
1	Points to a parameter address list that consists of two addresses: one address points to the general data and the other points to event-specific data. The layout of these areas is mapped by the DSECT RDCGDATA and RDCLDATA respectively. Both DSECTs are supplied in source form in the Natural macro NAMRDC.
13	Points to a 72-byte standard save area.
14	Contains the return address.
15	Contains either the entry point address or the return code of the user exit.

Note:

The user exits are called independently of the CMRDC interface.

A user-exit program must have the same attributes as Natural; that is, it must have the same addressing mode, and it has to be reentrant. It must be linked with the Natural nucleus according to the conventions of statically linked non-Natural programs. See also the profile parameters CSTATIC and RCA described in the *Parameter Reference* documentation.

By default, a 400-byte exclusive work area (per session) is supplied for each user exit (the field RDCGWRKA). If a larger work area is required for a user exit, it can be specified after the user-exit name in the Natural profile parameter RDCEXIT (see the relevant section in the *Parameter Reference* documentation). The work-area length is passed on to the user exits in the field RDCGWRKL and can be used for verification. The location of this work area can change during a session due to Natural relocation, but its contents are preserved.

In TP-monitor environments, the TP anchor address is supplied (the field RDCGANCH); for example, the CSA address under CICS. It can be used to access system information.

If a Program Check occurs during the execution of a user-exit program, further data collecting is disabled for the rest of the session to avoid recursive abend situations.

The section below contains information on:

- Return Codes
- Examples of User-Exit Programs

Return Codes

Non-zero return codes are only supported for two events:

- Before a database call, where Register 15 can contain an Adabas response code, which is stored in the control block; the Adabas call will not be executed.
- At program start, where Register 15 can contain a Natural error message code; the program will not be executed, but an error condition will occur with the specified number.

Examples of User-Exit Programs

The following examples of user-exit programs are provided in the following Natural source datasets:

Program	Dataset	Used for
NAMRDC	NAT nnn .SRCE	DSECT macro for general and event-specific data.
XNATRDC1	NAT nnn .SRCE	Natural TSO Interface and Natural batch interface under operating systems z/OS and z/VSE.
XNCFRDC1	NAT nnn .SRCE	Com-plete.
XNCIRDC1	NCI nnn .SRCE	Natural CICS Interface.

Calling the CMRDC Interface

The CMRDC application programming interface is used for retrieving and controlling the trace data recorded in the RDCSIZE buffer. This can be useful for testing purposes.

To invoke the CMRDC interface

- In the Natural program, issue a CALL statement.

The functions provided by the CMRDC interface and the syntax that applies to the corresponding Natural CALL statements are described in the following section.

In addition, the Natural system library SYSRDC provides example programs and the appropriate local data area RDCDATA.

The section below contains information on:

- Retrieving Trace Records
- Stopping and Restarting Trace Recording
- Selecting Event Types for Trace Recording
- User-Defined Events
- Retrieving the Trace Status
- CMRDC Return Codes
- Example Programs in Library SYSRDC

Retrieving Trace Records

To read the data from the RDCSIZE buffer, invoke CMRDC with the following Natural statement:

```
CALL 'CMRDC' function event-time gen-data event-data seq.-number
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible functions: F Get first trace record. G Get next trace record. N Get record of sequence number specified.
<i>event-time</i>	N10 or N12	Time of event. The time is passed back in either of the following formats: <i>HHMMSSXXXX</i> <i>HH</i> = hours, <i>MM</i> = minutes, <i>SS</i> = seconds, <i>XXXX</i> = fraction of a second <i>HHMMSSXXXXXX</i> <i>HH</i> = hours, <i>MM</i> = minutes, <i>SS</i> = seconds, <i>XXXXXX</i> = fraction of a second
<i>gen-data</i>	A252	General data.
<i>event-data</i>	A252	Event-specific data.
<i>seq.-number</i>	I4	Sequence number of record. Only applies to Function N (see above).

The retrieval functions stop recording data in the RDCSIZE buffer implicitly. To restart recording, use Function S described below.

Stopping and Restarting Trace Recording

To stop or restart trace recording data in the RDCSIZE buffer, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function
```

The following parameter is passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible functions: S Clear RDCSIZE buffer and restart trace recording. P Stop trace recording.

Selecting Event Types for Trace Recording

By default, all events are selected for trace recording. Use this function if you wish only specific events to be recorded.

Note:

This function only selects the events at which data is to be recorded in the RDCSIZE buffer. It does not affect the data passed to the user exits. Neither does it affect the status (started/stopped) of trace recording.

To select the types of event to be recorded, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function type . . .
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible function: T Select events for trace recording.
<i>type</i>	A1, A2 or A3	The one- or two-letter event code for the type of event to be recorded as listed in <i>Data-Collecting Events</i> . Specify any number of parameters for the event codes desired. Alternatively, you can select a range of events or no event: ALL All events. <i>value*</i> All events with event codes that start with <i>value</i> . For example, P* selects all events of the type program: PL, PS and PT. blank A blank character selects no event.

User-Defined Events

To specify a user-defined event, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function record
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible function: U User-defined trace event.
<i>record</i>	<i>Ann</i>	Trace record with a length (<i>nnn</i>) of up to 250 bytes.

Retrieving the Trace Status

To retrieve the trace status, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function status types
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible function: C Retrieve trace status.
<i>status</i>	A1	Current trace status: S Trace active. P Trace inactive.
<i>types</i>	<i>Ann</i>	The two-letter event codes of the event types currently set; see <i>Selecting Event Types for Trace Recording</i> . The variable <i>types</i> requires a minimum length (<i>nn</i>) of 64 bytes to provide sufficient space for all event types. This parameter is optional.

CMRDC Return Codes

Code	Meaning
0	Function executed successfully.
4	Last trace record Only applies to Functions F and G (see above).
8	Too few parameters for this function.
12	Invalid function code.
16	Natural Data Collector not active, for example, RDCSIZE=0.
20	Natural Data Collector disabled after an error.
24	No buffer space available for trace recording (RDCSIZE=2 or smaller).
28	Invalid parameter value. Only applies to Functions T and N (see above).

Example Programs in Library SYSRDC

The Natural system library SYSRDC contains the following example programs:

Program	Function
RDCDISP	Display all records in the RDCSIZE buffer and show the fields as specified in the program. See also <i>Example Output of Program RDCDISP</i> below.
RDCSTART	Restart trace recording.
RDCSTOP	Stop trace recording.
RDCSET	Select events for trace recording.
RDCUSER	User-defined event.
RDCSTAT	Retrieve trace status.

Example Output of Program RDCDISP

The example screen below shows the extract of an output report produced by the example program RDCDISP.

ETIME1	TY	GCUID	LV	GPGM	T	GCAPL	PRLIB	PRNAM	CO	DB	FN	RC
0.0000	SI	SAG				SYSTEM						
0.0002	DB	SAG				SYSTEM			OP	10		
0.0002	DA	SAG				SYSTEM			OP	10		
0.0000	DB	SAG				SYSTEM			S1	10	1640	
0.0005	DA	SAG				SYSTEM			S1	10	1640	
0.0001	PL	SAG				SYSTEM	SYSLIB	MAINMENU				
0.0001	PS	SAG	1	MAINMENU	F	SYSTEM	SYSLIB	MAINMENU		10	1640	
0.0001	PL	SAG	1	MAINMENU		SYSTEM	SYSLIB	NAT00029				
0.0000	DB	SAG	1	MAINMENU		SYSTEM			S1	10	1640	
0.0002	DA	SAG	1	MAINMENU		SYSTEM			S1	10	1640	
0.0000	PS	SAG	2	NAT00029	N	SYSTEM	SYSLIBS	NAT00029		10	1640	
0.0000	CB	SAG	2	NAT00029		SYSTEM		CMMPP	S			
0.0000	CA	SAG	2	NAT00029		SYSTEM		CMMPP	S			
0.0000	PT	SAG	2	NAT00029	N	SYSTEM	SYSLIBS	NAT00029				
0.0000	PL	SAG	1	MAINMENU		SYSTEM	SYSLIB	USR2003P				
0.0000	PS	SAG	2	USR2003P	N	SYSTEM	SYSLIB	USR2003P		10	1640	
0.0001	PT	SAG	2	USR2003P	N	SYSTEM	SYSLIB	USR2003P				
0.0000	CB	SAG	1	MAINMENU		SYSTEM		CMUB	S			

The following table describes the columns that are displayed on the example screen, and the variables to which they refer. For further information, see the comments in the program source and the local data area RDCDATA. For an explanation of the Natural system variables mentioned, refer to the *System Variables* documentation.

Column	Explanation
ETIME1	Time interval in seconds between the execution of the current and the previous event.
TY	Type of event as listed in <i>Data-Collecting Events</i> .
GCUID	Current Natural user ID as assigned by the Natural system variable *USER.
LV	Program level.
GPGM	Name of the current program as assigned by the Natural system variable *PROGRAM.
T	Type of program.
GCAPL	ID of the current application library as assigned by the Natural system variable *APPLIC-ID.
PRLIB	ID of the library where the program is stored. Applies to events of the type program, for example, Event Code PL.
PRNAM	Name of the program to be loaded for the type of event.
CO	Database command.
DB	Database ID.
FN	File number of Database.
RC	Database response code.