

# SYSRDC Utility

The utility SYSRDC collects data at certain events within Natural. These data can be used for the following purposes:

- Via exit points, the data can be passed to external monitoring and accounting programs for the evaluation of activities in Natural sessions.
- Via a CALL interface, the data can be evaluated in a Natural program to obtain information about the execution flow of a Natural application in the current Natural session. This may be useful for testing purposes.

The data are always supplied at the exit points provided by SYSRDC to be used by external monitoring/accounting programs.

Optionally, the data can also be written into a special SYSRDC buffer. SYSRDC supplies a CALL interface to retrieve the contents of this buffer. With this interface, you can also start and stop the writing of data into the SYSRDC buffer, select events for which data are to be recorded, and write your own events. In this section, the writing of event data into this buffer is referred to as trace recording.

This section covers the following topics:

- Overview
  - Exit Points for External Monitoring/Accounting Programs
  - Trace Recording - CALL Interface
-

## Overview

- Data-Collecting Events
- Collected Data
- Activating SYSRDC

## Data-Collecting Events

SYSRDC collects data at the following events within Natural (the letters in parentheses are the corresponding event codes used by SYSRDC's CALL interface):

- 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 (N), \*
- at a user-supplied event (U).

\* To activate the "internal trace call" event, the profile parameter ITRACE=ON has to be specified. The Natural components that are to issue internal trace calls have to be defined in the NTTRACE macro of the Natural parameter module (or with the corresponding dynamic profile parameter TRACE).

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

## Collected Data

The collected data can be split in two categories:

- **general data** - which are collected at every event;
- **event-specific data** - which are only collected at specific events.

For the layout of the data, see the macro NAMRDC in the source library, or the local data area RDCDATA in library SYSRDC.

## General Data

The following data are collected at every event:

- Software AG product name,
- product version,
- operating system,
- TP monitor,
- run modes (OS/390 systems),
- TP user or batch job name,
- TP terminal ID,
- current Natural user ID,
- current Natural Security user group ID,
- current library,
- current Natural program,
- current program level,
- line number of currently executed statement.

## Event-Specific Data

The following data are only collected at the following events:

Event	Data Elements
Session Initialization	none
Session Termination	termination return code termination message ID name of back-end program
Program Load	name of program to be loaded name of load library
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 parameter list pointer
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 called program calling mode program link location parameter type parameter address program entry address response code
Runtime Error	error number external abend code error handling program
Internal Trace	up to 250 bytes of information
User-Supplied Event	up to 250 bytes of information

## Activating SYSRDC

The SYSRDC utility is activated and controlled by the Natural profile parameter RDCSIZE.

This parameter also determines the size (in KB) of the SYSRDC buffer.

By default, the parameter is set to RDCSIZE=0, which means that the SYSRDC utility is deactivated altogether.

If you set RDCSIZE=2, SYSRDC is activated. The collected data are supplied to the exit points, but are not written into the SYSRDC buffer.

If you wish to activate trace recording (that is, if you wish the data also to be written into the SYSRDC buffer in order to retrieve them within your Natural application via the supplied CALL interface), you have to set the RDCSIZE parameter to a value greater than 2.

To calculate the number of records that fit into SYSRDC buffer, you use the following formula:

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

The maximum size of the buffer is 128 KB.

## Exit Points for External Monitoring/Accounting Programs

SYSRDC supports any number of concurrent exit points. An exit point can be defined by profile parameter RDCEXIT (see the Natural Parameter Reference documentation). An external monitoring/accounting program can be attached to each exit point.

At every event listed under Data-Collecting Events, the exit points obtain control, using standard linkage call conventions as follows:

Register	Content
1	Points to a parameter address list, consisting of 2 addresses pointing to the general and event-specific data. The layout of these areas is mapped by the DSECT RDCGDATA and RDCLDATA respectively. Both DSECTs are supplied in source form by 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 exit.

**Note:**

The exits are called independently from the internal CALL interface.

An 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 (profile parameters CSTATIC or RCA, as described in the Natural Parameter Reference documentation).

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

In teleprocessing environments, the TP anchor address is supplied (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 SYSRDC or of an exit program, further data collecting is disabled for the rest of the session to avoid recursive abend situations.

## 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 into the control block; the Adabas call will not be executed.
- At program start, where Register 15 can contain a Natural error message number; the program will not be executed but an error condition will occur with the specified number.

## Sample Exit Programs

The following sample accounting exit programs are provided in the following source libraries:

<b>Program</b>	<b>Contained in Source Library</b>	<b>Contents</b>
NAMRDC	NATSRCE	DSECT macro for general and event-specific data.
XNATRDC1	NATSRCE	Sample exit program for TSO and OS/390 or VSE/ESA for batch mode.
XNCFRDC1	NATSRCE	Sample exit program for Com-plete.
XNCIRDC1	NCISRCE	Sample exit program for CICS.

## Trace Recording - CALL Interface

Trace recording is activated with the profile parameter RDCSIZE and can be used to collect the event data of the current Natural session in a special incore SYSRDC buffer for testing purposes, without using the exits described earlier.

The size of this buffer is determined by the RDCSIZE parameter. The 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 buffer content is deleted.

Trace recording is started automatically during session initialization and all events are recorded.

The data in the buffer can be accessed by any Natural program within the same session.

Several sample programs, as well as a local data area for layout of the data (RDCDATA), are provided in the library SYSRDC.

Following is information on:

- Program CMRDC
- CMRDC Functions
- CMRDC Return Codes
- Sample Programs in Library SYSRDC

### Program CMRDC

The communication between a Natural program and trace recording is performed by invoking the program CMRDC with CALL statements. CMRDC provides the following functions:

- retrieval of trace records,
- start and stop of trace recording,
- selection of event types for trace recording,
- writing of data at user-supplied events.

## CMRDC Functions

### Retrieval of Trace Records

To read the data from the SYSRDC buffer, you invoke CMRDC with the following statement:

```
CALL 'CMRDC' function event-time gen-data event-data
```

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.
<i>event-time</i>	N10	Time of event ( <i>HHMMSSXXXX</i> ).
<i>gen-data</i>	A252	General data.
<i>event-data</i>	A252	Event-specific data.

The retrieval functions automatically stop the writing of data into the SYSRDC buffer. To start it again, you use the CALL statement described below.

## Start and Stop of Trace Recording

To start or stop trace recording (that is, the writing of data into the SYSRDC buffer,) you 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 SYSRDC buffer and start trace recording.  P Stop trace recording.

## Select Events for Trace Recording

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

**Note:**

This function only selects the events at which data are to be written into the SYSRDC buffer. It does not affect the data passed to the exits. It does not affect the status (started/stopped) of trace recording, either.

To select the types of events to be recorded, you 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	Type of event to be recorded:  PL Program load PS Program start PT Program termination DB Before database call DA After database call CB Before non-Natural program call CA After non-Natural program call IB Before terminal I/O IA After terminal I/O E Runtime error N Internal trace call NS Statement trace U User trace call  ALL All events ' ' ( <i>blank</i> ) No event

To select multiple events that begin with the same character, you can use asterisk notation. For example, with **P\*** you would select all program events (PL, PS and PT).

## Write User-Defined Trace Entry

To write a user-defined event into the SYSRDC buffer from within a program, you 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 Write user-defined trace entry.
<i>record</i>	Annn	Trace record with a length ( <i>nnn</i> ) of up to 250 bytes.

## CMRDC Return Codes

Code	Meaning
0	Function successfully executed.
4	Last trace record (Functions <b>F</b> and <b>G</b> only).
8	Too few parameters for this function.
12	Invalid function code.
16	SYSRDC not active (for example, RDCSIZE=0).
20	SYSRDC disabled after an error.
24	No buffer space available for trace recording (RDCSIZE=2 or smaller; Function <b>U</b> only).
28	Invalid parameter value (Function <b>T</b> only).

## Sample Programs in Library SYSRDC

The library SYSRDC contains the following sample programs:

Program	Function
RDCDISP	Sample program to display all records in the SYSRDC buffer showing some selected fields.
RDCSTART	Sample program to start trace recording.
RDCSTOP	Sample program to stop trace recording.
RDCSET	Sample program to select events for trace recording.
RDCUSER	Sample program to write user trace entries.