



DB2RCF

DB2 Resources Checking Facility

The missing link for the permanent availability monitoring and checking of z/OS DB2 database objects

Since several years DB2RCF allows UBS AG, the largest Bank in Switzerland, to successfully monitor and efficiently control the availability of its huge amount of mission critical DB2 tables in multiple and complex parallel sysplex data-sharing and SAP/R3 environments.

- ✓ Improves the overall availability of DB2 objects
- ✓ Permits to prevent and to avoid '-904' - abends
- ✓ Allows to plan the unavailability of specific tables
- ✓ Informs users and applications about the unavailability of tables as well as the reasons of an unavailability
- ✓ Allows to be informed about the impact of unavailable tables (eg. which plans or packages cannot be executed)
- ✓ Provides useful information for service monitoring and allows to comply with SLA's

What does it offer?

The main purpose of *DB2RCF* is to provide functions which allow to:

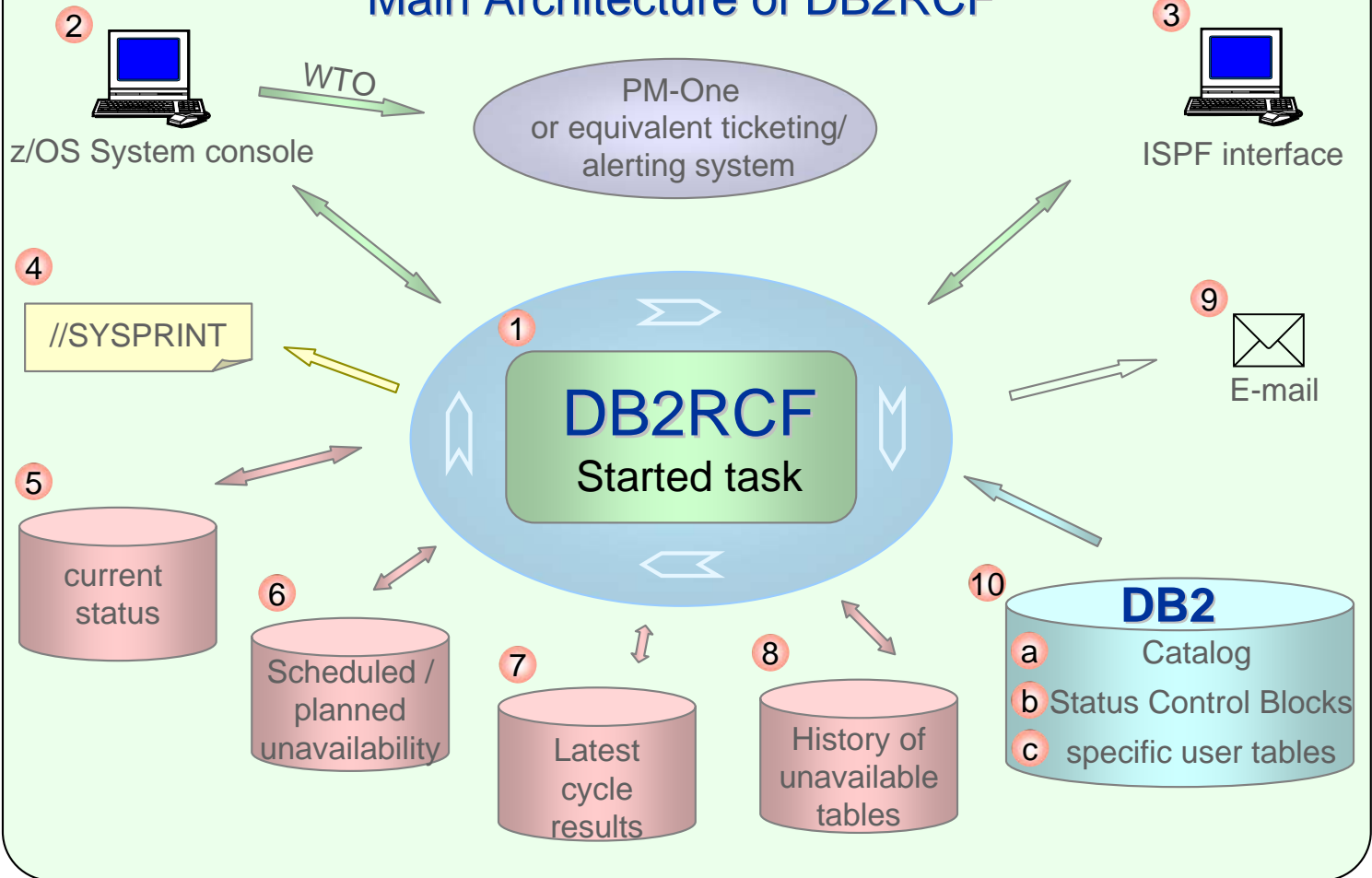
- perform an online checking of the availability of specific DB2 tables
- inform (users and/or applications) about unavailable DB2 tables and the reason of the unavailability

While periodically checking (e.g. every 5 min.) the DB2 tables availability, latent restrictions at tablespace and indexspace level (COPYP, RECP, CHKP,...) as well as lost connections (DDF/DRDA), stopped utilities etc. will be detected additionally.

What are the main features offered by DB2RCF to guarantee an efficient availability control ?

- detects all DB2 databases which have restrictions (stopped, excl. locked, ...)
- detects all DB2 tablespaces and indexspaces which have restrictions (COPY, CHKP, RECP, LPL, ...)
- detects all DB2 DDF/DRDA locations which have a lost connection
- detects all DB2 DRDA locations inactive since a defined amount of time
- detects all stopped and incomplete utilities
- detects all threads inactive since a defined amount of time
- detects all DB2 locks without an associated thread
- detects DB2 tables which have been dropped
- checks if a specific DB2 table can be read or updated (read/write trial)
- allows to schedule status changes for specific objects
- informs users and applications about the detected unavailable DB2 resources with WTO's, SYSPRINT messages, e-mails and entries in *DB2RCF* controlled DB2 tables via the *DB2RCF* ISPF Interface
- Allows applications to query (via SQL) the *DB2RCF* status table in order to obtain an availability status (read-write, read only, stopped, scheduled stopped or dropped) instead of obtaining an error code '-904'

Main Architecture of DB2RCF



- 1 *DB2RCF* MVS Started Task, which provides DB2 tables with the status of specified objects and checks and updates their current status.
- 2 z/OS System console, where the operator can enter commands, send and receive messages.
- 3 ISPF Interface, where a user can execute *DB2RCF* administrative and status functions via ISPF panels.
- 4 Output from the *DB2RCF* Started Task.
- 5 DB2 table containing the current status of specified DB2 tables.
- 6 DB2 table for scheduling planned availability status changes in the future.
- 7 DB2 table holding the result of the last monitoring cycle.
- 8 DB2 table containing the history of unavailable resources.
- 9 Possibility to inform a user about an unavailable table he is subscribed for.
- 10a DB2 Catalog with the information about all DB2 objects of a subsystem.
- 10b The status of DB2 objects are written in the DB2 Status Control Blocks and obtained via DB2 DISPLAY commands.
- 10c In the 5 (current status table) defined user tables which should be monitored by the *DB2RCF* Started Task.

Who will Use DB2RCF

Application Developers	✓
DBA's	✓
Production Planners	✓
Operations	✓
Technical Support	✓
Systems Engineers	✓

DB2RCF is a member of the KeyTools product family. KeyTools and *DB2RCF* are trademarks or registered trademarks of UFD Software AG.

Prerequisites

DB2RCF supports all current versions of DB2 UDB for OS/390 and z/OS.