INTERCHIP

Real-Time Standards (RTS) Version 4.10

General Information Manual

This Manual introduces the highlights of RTS, and provides an overview of the functionality which RTS offers.

The technical level of the information contained in this Manual is deliberately kept quite low. It is intended to be easily readable by non-technical personnel.

RTS's functionality: a short summary

RTS's functionality has already been summarised, in the Preface, in one phrase:

enforcement of installation-controlled, real-time adjustable installation standards

The following expands this summary a little, and introduces the chief features and facilities of RTS. But first, a brief summary of the need for standards enforcement would be helpful.

MVS offers an immensely wide range of services and options. It is no great overstatement to say that just about everything is there which just about anybody might need.

Unfortunately, the real resources which your installation can offer are far more limited. With maybe hundreds of thousands of batch jobs, on-line users, on-line transactions and the like, there will be at best chaos, and at worst death by strangulation, if everybody is free to use as much as he wants of whatever he wants, whenever he wants.

Every installation must direct and limit its users - "... you can't use this, you can't have so much of that ...". Installation standards are required - and people being what they are, enforcement is also required!

RTS addresses these issues through the following features and facilities.

° **Ready-to-run standards enforcement functions**

  RTS incorporates a wide range of standards enforcement functions, "ready-to-run".

  MVS offers a broad range of customisation possibilities, but these possibilities are only a base; actual functionality must be provided from external sources.

  RTS provides the actual standards enforcement functionality and, as will be seen shortly, the range provided by RTS is far more extensive than that which typically exists at MVS installations.

° **Unparalleled processing control facilities**

  RTS provides real-time administrative control over all aspects of standards enforcement, through highly flexible facilities which allow you to control the scope of its processing. For example, you may have RTS

  - process all jobs, or just selected jobs which you specify,
  - vary, from one job to another and one time to another, the actions taken by RTS, and
  - vary, from job to job and from time to time, whether RTS should perform real processing, or just simulation (e.g., for learning or evaluation purposes)

  all fully automatically, on the basis of specifications which you supply.

Furthermore, there is ISPF panel support for the creation and maintenance of all your RTS specifications, and for the displaying of information about the standards enforcement actions actually taken by RTS.

And naturally, the on-line nature of this ISPF support allows you to adjust your standards real-time. Simply update your RTS specifications on-line, and your systems will automatically begin to enforce your adjusted standards.

RTS’s special advantages

The two possibilities: RTS or local development

As already noted, MVS offers a wide range of customisation possibilities. There are probably very few installations which have not already used these possibilities as the basis for the local development of some standards enforcement functions.

But the disadvantages for your installation of such a local approach are not difficult to see.

° **You get to be permanently involved in local development**

  You have to invest not only in initial development, but also in further development as your environment changes, and in maintenance, and in documentation, ...

° **You get only a local subset of standards enforcement functions**

  Naturally, your range of functions will be based on your local implementation techniques and local requirements, and is unlikely to include more than a subset of the RTS range of functions. Indeed, your current local subset is probably limited not simply to your current requirements, but to your most urgent current requirements!

° **You get only elementary capability**

  If yours is a typical installation, then your standards enforcement might be relatively unsophisticated, at least in its implementation, with numerous built-in assumptions and conditions which reflect your current local environment and requirements. Such built-in assumptions and conditions are both difficult and time-consuming to change, which reintroduces the issue of further local development ...

On each of these counts, RTS offers advantages.

° **RTS delivers ready-to-run functionality**

  You can profit from the cost-effectiveness of ready-to-run software, by eliminating your need to invest in continual local development, with all its associated overheads, limitations and personnel requirements.

° **RTS delivers a wide range of standards enforcement functions**

  You can benefit from a far wider scope of functionality than any local development is ever likely to offer.

° **RTS delivers sophisticated capability**

  You can enjoy RTS’s various features and facilities for improving the effectiveness of your standards enforcement:

  - Use the precision of RTS’s highly flexible and granular specification capabilities to make your standards enforcement definitions reflect your requirements - not just approximately, but exactly.
  - Use the convenience of RTS’s real-time administrative control facilities whenever your environment or requirements change, to adjust your standards enforcement definitions on-line, and within seconds, with no disruption at all, your systems are enforcing your new requirements.

- If you are ever in any doubt about what effects your adjustments would actually have, utilise the protection of RTS's full simulation facilities to see what would happen, before you make any real changes.
- Then, whether you were simulating or not, profit from the assurance provided by RTS's full logging and on-line log viewing facilities to verify the correctness of your changes.

Cost-effectiveness: get more from your budget

As can be seen clearly from the preceding comparison of RTS versus local development, the first advantage of using RTS is that of cost-effectiveness: how much more effective a powerful product like RTS can be, at a cost which generally compares more than favourably with the cost of locally-developed functionality (not to mention the hidden costs of having ineffectively-controlled system resources).

Each installation clearly has to evaluate the cost-effectiveness issue for itself, but it will be a rare installation which is unable to make substantial direct and/or indirect savings.

Scope: a wide range of standards enforcement functions

The next advantage is that of scope: the sheer range of functionality which RTS offers. The scope of RTS includes:

- JOB-Level control
- STEP-Level control
- DD-Level control

Assurance: logging and log viewing facilities

RTS's logging and log viewing facilities provide the assurance which is another practical necessity with such a powerful product.

RTS is able to log every single standards enforcement action (including simulated actions) which it takes, in either, or both, of two forms.

- User-directed logging: messages to job logs or TSO sessions.

  Each user-directed copy of a message is sent to the individual job log of the particular job (or TSO user) to which the message applies.

  The objective is to inform the appropriate user, directly, about individual standards enforcement actions which RTS has taken in respect of his particular job .
- Administrator-directed logging: SMF records.

  Each administrator-directed copy of a message is sent to the system's central SMF datasets - MVS's standard, global, machine-readable collection point for general system activity tracking information.

  The objective is to inform the RTS administrator, globally, about all standards enforcement actions which RTS has taken in respect of any job.

The RTS log viewing facilities are uncomplicated and convenient to use. You simply

- call up the on-line TSO/ISPF component of RTS, and
- select the "Log information" option,

and RTS provides direct access to the stored and sorted log information.

RTS processing flow and RTS datasets

Having introduced the main features and facilities of RTS, we will complete our introductory look at RTS by describing how they fit together.

This involves looking at the processing flow within RTS, and the datasets which RTS uses.

RTS processing flow

RTS processing is divided between three components: the specification entry component, the main execution component, and the log extract / viewing component.

The RTS specification entry component

The RTS specification entry component provides the facilities which allow you to enter details of your RTS requirements.

You specify your requirements through two levels of RTS specifications - RTS general control specifications, and RTS function control specifications.

You enter your RTS specifications in TSO foreground, through ISPF panels supplied by RTS. The panels guide you through a logical sequence of actions, in which you enter your requirements in human-readable form.

When you have finished entering your requirements, you use the ISPF panels to have your requirements translated from this human-readable form into the RTS-internal format needed by the RTS main execution component.

The RTS main execution component

The RTS main execution component reads the RTS-internal format version of your requirements, and creates a working copy for itself.

The RTS main execution component operates in a background environment, within the address space of the RTS started task - a permanently-running task, which is (typically) started up automatically at each system restart. At intervals which you specify (in your RTS general control specifications), the RTS main execution component checks your specifications (in case you have just updated them), and refreshes its working copy if so.

Actual processing by the RTS main execution component is asynchronous. MVS handles a continuous stream of user jobs, TSO sessions, on-line transactions and the like; whenever any one of these reaches any standards enforcement point, the RTS main execution component is called.

At each such call, the RTS main execution component analyses its working copy of your specifications, and, based on what it finds there, makes its standards enforcement decision, and performs any specified standards enforcement action on the job. It then allows job flow for that job to continue, and waits for the next standards enforcement call.

The RTS main execution component writes SMF records as required, which describe exactly what standards enforcement actions it has performed. These SMF records are the input to the RTS log extract / viewing component.

The RTS log extract / viewing component

The RTS log extract / viewing component is really two components - the RTS log extract component and the RTS log viewing component.

- The RTS log extract component allows you to extract the RTS SMF records from the SMF datasets, merge and sort the results, and store the complete set in the form required by the RTS log viewing component.
- The RTS log viewing component allows you to request displays about the standards enforcement actions taken by RTS. Once again, you do this in

TSO foreground, through the ISPF panels supplied by RTS. The ISPF panels guide you to the displays you require.

Note that although the RTS main execution component is processing continually, you are allowed at any time to update your RTS specifications, or to request displays of the standards enforcement actions taken by RTS. Each time you update your RTS specifications, the RTS main execution component will automatically switch to your new specifications. Similarly, each time you run the RTS log extract component, it will extract the latest RTS SMF records from the SMF datasets; correspondingly, each time you request a display about the standards enforcement actions taken by RTS, the RTS log viewing component will display the latest extracted RTS SMF records.

The key to understanding how all this works lies in the RTS execution datasets.

The RTS execution datasets

The RTS execution datasets are the basis of communication between the various RTS processing components. Figure 1 illustrates the RTS processing components, the RTS execution datasets, and the relationships between them.

As can be seen, two main RTS datasets are involved - the RTSPARM dataset, and the RTSCDS dataset. In addition, MVS’s SMF datasets are also involved.

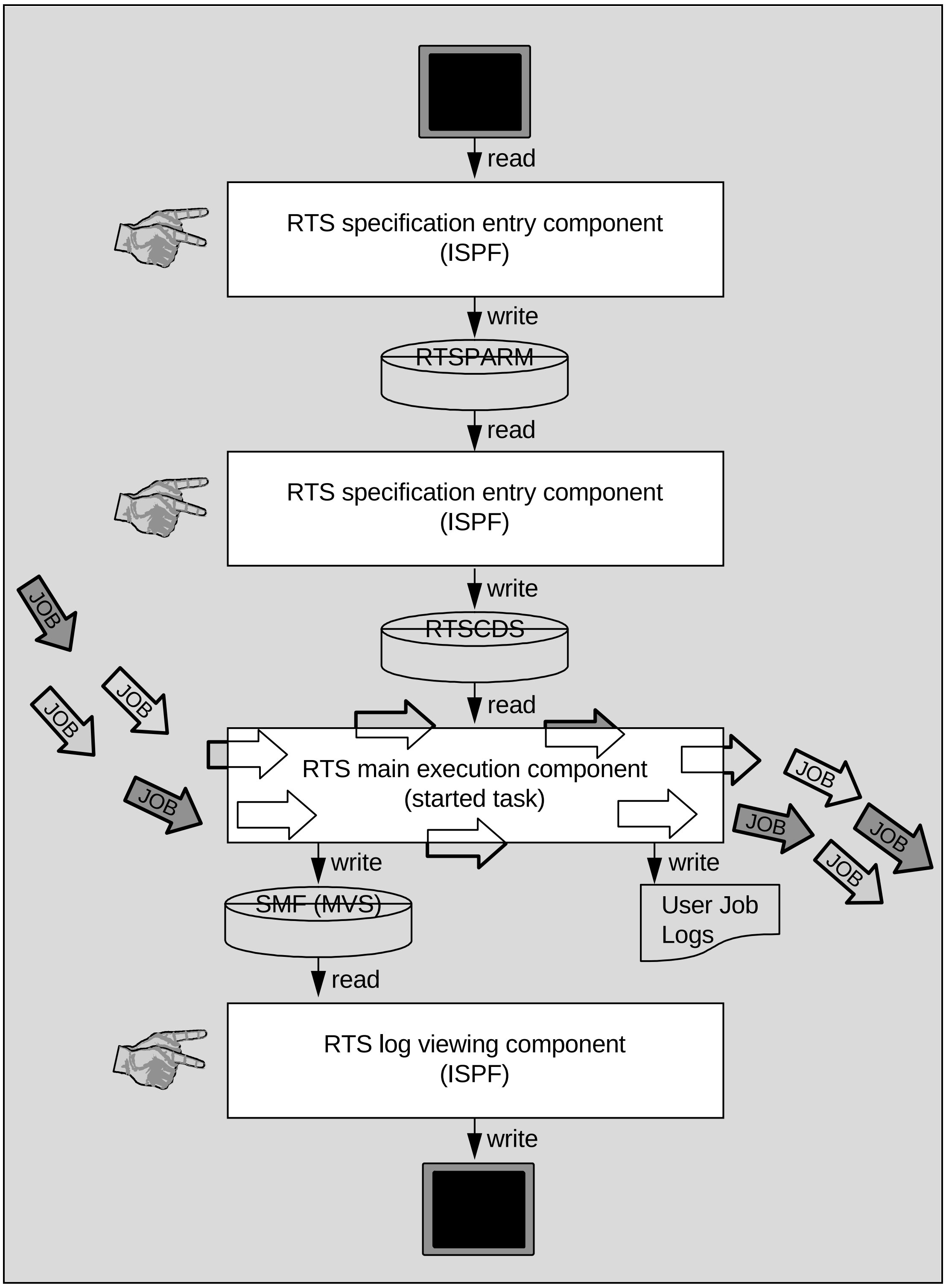
read
RTS specification entry component (ISPF)
write
RTSPARM
read
RTS specification entry component (ISPF)
write
RTSCDS
read
JOB
RTS main execution component (started task)
write
SMF (MVS)
write
User Job Logs
read
RTS log viewing component (ISPF)
write

Figure 1: RTS's processing components and execution datasets

Initially, when you use RTS's ISPF panels to enter your RTS specifications in human-readable form, RTS writes your RTS specifications (in exactly that form) to your RTS specification library (RTSPARM). This may be any ordinary card-image PDS[1] - it is up to you whether or not you use a dedicated PDS.

When you use RTS's ISPF panels to translate your specifications, the translation process reads your specifications from RTSPARM, and writes them to the RTS

[1] or PDSE

execution control dataset (RTSCDS) - a VSAM Linear Dataset, which is the primary dataset used by RTS execution.

If you update your RTS specifications, you use RTS's ISPF panels in exactly the same way, and the translation process writes your updated specifications to the RTSCDS. Assuming that RTS is running from the RTSCDS to which the translation process writes its output, then your changes become effective automatically. This is because the RTS started task continually reviews the RTSCDS, precisely in order to detect RTS specification changes.

During execution, RTS records its standards enforcement actions by writing SMF records to MVS's SMF datasets.

The RTS base datasets

As well as the RTS execution datasets, RTS has three further datasets - the RTSLOAD dataset, the RTSISPF dataset, and the RTSSAMP dataset - known as the RTS base datasets.

- The RTSLOAD dataset contains the RTS load modules.
- The RTSISPF dataset contains all the RTS ISPF definitions - ISPF panel definitions, ISPF message definitions, and so on.
- The RTSSAMP dataset contains sample RTS installation jobs and data members.

The essential difference between the RTS base datasets and the RTS execution datasets is that the content of the RTS base datasets is static, and comes from the RTS distribution tape, whereas the content of the RTS execution datasets is dynamic, and comes from RTS processing.

Summary

To sum up, RTS offers the following features and facilities:

Standards enforcement functions
An incomparable range

Highly flexible specification capabilities
Precision
Granularity

Real-time administrative control facilities
On-line
Real-time
Non-disruptive

Powerful simulation facilities
Precision and granularity inherited from specification capabilities

Logging and log viewing facilities
Precision and granularity inherited from specification capabilities

This completes the introductory overview of RTS. For those who need to know more, the remaining product manuals provide comprehensive information.