

Introduction

This chapter contains a brief overview of the Software AG Multiple Architecture Runtime System (SMARTS) and the requirements to run it.

This chapter covers the following topics:

- The SMARTS Runtime Environment
-

The SMARTS Runtime Environment

SMARTS supports multiple platforms and environments on each platform. It implements a standard runtime environment and supports standard C library functions such as standard I/O, sockets, pthreads, and semaphores.

Although primarily a C runtime environment, SMARTS extensions provide access as well to the facilities from other programming languages including Assembler, COBOL, and PL/1.

SMARTS operates in two distinct environments: server and client.

The SMARTS Server Environment

SMARTS supports a high performance environment for running server components on the different supported platforms: OS/390, VSE/ESA, VM/GCS, MSP (FACOM), and BS2000.

The SMARTS server environment provides a multitasking architecture that uses blocks of storage called 'threads' in which SMARTS application programs can run. By ensuring that all application storage is in contiguous blocks of the address space, the server environment is in a position to move dormant applications out of the address space to make room for more active users. If or when such a dormant application becomes active again, it is simply moved back into the address space and dispatched again.

The SMARTS server environment

- uses its multitasking structure to make full use of operating system subtasks to drive the system CPUs.
- uses contiguous storage threads to give it more control over the applications' storage areas, thus enabling it to handle more applications running in the one address space.
- shares the underlying operating system subtasks between users and is thus not subject to restrictions on the number of processes that can be concurrently supported.
- uses a state-of-the-art buffer pool management technique that ensures consistent path lengths, no fragmentation of storage areas, and expansion and contraction of the storage areas within the address space.
- uses a state-of-the-art resource manager that ensures the shortest path length possible for the serialization of resources. The technique uses only machine instructions unless it is necessary to wait for a resource.

- is ready now for 24-by-7 operation as the nucleus does not need to be cycled for any internal reasons and the buffer manager and resource manager can handle higher than expected loads subject to the resources being available in the address space.

The SMARTS Client Environment

On the supported platforms, SMARTS-based applications are supported in a number of so-called client environments: batch, OS/Transaction Server (CICS), Com-plete (version 6.1 or above), and UTM. In all cases, the support enables client applications to use the SMARTS SDK library functions:

Server Environment	Supported Client Environments
OS/390	Batch, TSO, OS/TS, Com-plete
VSE/ESA	Batch, OS/TS, Com-plete
MSP (FACOM)	Batch, Com-plete
VM/GCS	Batch (command line applications only)
BS2000	Batch/TIAM, UTM

In general, the client environments support everything that the server environment supports; however, Software AG recommends that you avoid running heavy duty applications in client environments. For example, a 'pthreaded' application should not be run in an OS/TS environment, even though it is functionally possible.

Supported TCP/IP Stacks

- IBM TCP/IP version 3.1 or above.
- IBM TCP/IP version 3.1 with PTF UN93769 and all its pre- and co-requisite PTFs applied.
- IBM TCP/IP version 3.2 with PTF UN99683 and all its pre- and co-requisite PTFs applied. If PTF UQ09354 is applied, PTF UQ11919 must also be applied; otherwise, the system hangs.
- Interlink TCP/IP version 3.1 or above.
- VSE/ESA version 2.2 or above using the Connectivity Systems TCP/IP 4 VSE stack.
- MSP/EX from Fujitsu using the TISP TCP/IP stack.
- VM/ESA using the standard IBM stack available on VM.