HOT TOPICS

A z/OS Newsletter

In this issue...

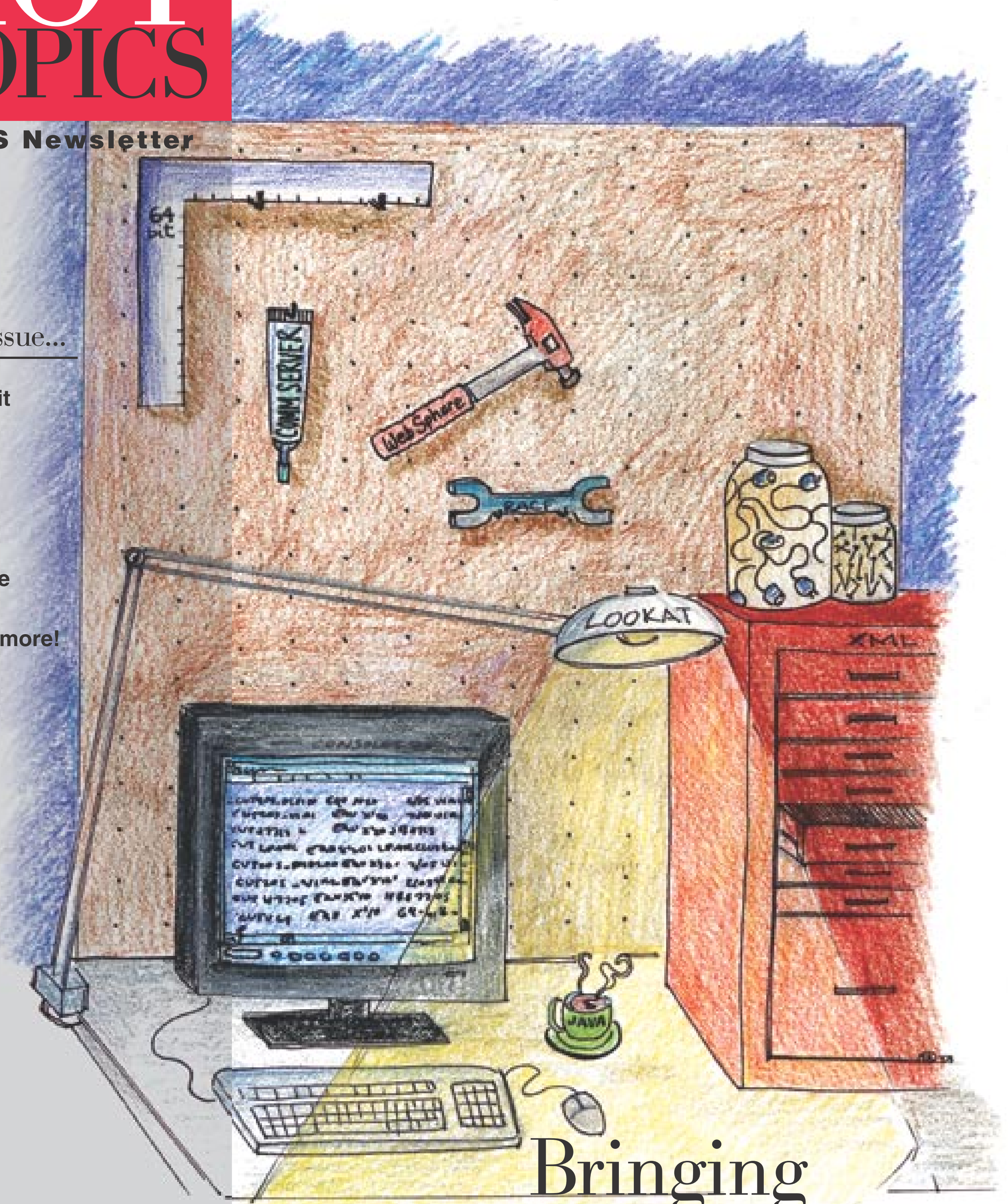
XML Toolkit

64-bit

Migration

WebSphere

and much more!



Bringing the tools to you

August 2003 - Issue 9

On the up and up! System Automation for OS/390 offers a high availability solution for WebSphere for z/OS

BY ULRICH KETTNER

There are many advantages to using system automation. One is the reduction in operating costs you may see after you simplify your system management. You will want to do this when running a WebSphere Application Server on z/OS because:

- WebSphere for z/OS is based on proven components of z/OS and therefore interfaces with a considerable number of products.
- WebSphere for z/OS uses many z/OS address spaces to deliver scalable and high performance Web services.

This increased operational complexity makes it worth while to set up automation with the objective of simplifying operations.

Another advantage to using system automation is the increased application availability you get from setting your applications to restart automatically or move from a failing image to a healthy image within a sysplex. This makes good business sense because, when you run Internet applications, the availability of the hosting system is exposed to the public at all times. It is therefore essential to provide a high availability setup for systems that provide Internet services such as with WebSphere for z/OS.

IBM has developed a high availability solution for WebSphere Application Server V4.0.1 for z/OS and OS/390 (WebSphere for z/OS) based on System Automation for OS/390 V2.2 (SA for OS/390).

What's available?

With SA for OS/390 V2.2, IBM supports a fully automated operational environment for all components, prerequisites and related products of WebSphere Application Server V4.0.1 for z/OS and OS/390. APAR OA02375 for SA for OS/390 V2.2 provides:

- An enhanced message table including messages issued by WebSphere for z/OS and related products during startup and termination.
- A common routine used to clean up server address spaces that WebSphere for z/OS is unable to terminate during its end processing.

The following information and downloads are available on the System Automation Web site at http://www.ibm.com/servers/eserver/zseries/software/sa/:

- A white paper you can use as a guideline.
- A sample policy database (PDB) that contains the definitions used to set up the environment described in the white paper. You can use this PDB as a sample to set up SA for OS/390 V2.2 for your WebSphere for z/OS installation.
- Sample tools for automating WebSphere for z/OS administration.

What's automated?

WebSphere for z/OS uses z/OS components that provide a stable platform with the following built-in recovery and security functions:

- TCP/IP as a communications vehicle
- UNIX System Services platform support with HFS for file support
- LDAP and DB2 for directory services and database support
- WLM for workload management
- RACF for security
- RRS for recovery.

These software components (or equivalent products) are prerequisites for WebSphere for z/OS. In addition, WebSphere for z/OS uses IBM HTTP Server for z/OS as the Web server.

The automation setup ensures that all basic system components, such as JES2, RRS, and TSO, and the WebSphere for z/OS prerequisites are brought up on all required systems in the right sequence. These components are monitored, brought down, restarted, and moved to alternate systems if required.

WebSphere Application Server V4.0.1 for z/OS and OS/390 consists of a base environment and application servers. The base environment consists of a daemon that in turn brings up everything else in the base environment:

- Systems Management Server (SMS) that controls the WebSphere for z/OS configuration
- Naming server that locates Java objects such as EJBs
- Interface Repository (IR) server, which manages CORBA business object interfaces.

The application servers consist of the following:

- J2EE servers that provide run-time environments for J2EE components such as servlets or EJBs. They support only Java.
- CORBA (MOFW) servers (for CORBA business objects), which support various different languages.

Each of these servers (except the daemon) consists of a control region and multiple server regions that WLM starts as needed.

SA for OS/390 automatically brings up the WebSphere for z/OS base environment on all sysplex images and considers it part of the infrastructure. It also automates two J2EE servers. Instances of each server are started on two system images. SA for OS/390 knows about and controls the control regions only; server regions are not defined as SA for OS/390

continued on page 29

continued from page 24

objects. The sysplex consists of three systems. Figure 1 shows a possible layout.

What makes WebSphere for z/OS special?

Three aspects specific to WebSphere for z/OS impact the automation setup: the control region and server region architecture, the WebSphere for z/OS administration procedure, and the HTTP session concept.

The control region and server region architecture

SMS, Naming and IR control regions, as well as J2EE or CORBA control regions, all start server regions as needed to satisfy the service goals defined by WLM. Control regions are defined in and handled by SA for OS/390. Server regions are not monitored by SA for OS/390. All server regions started by a control region have the same jobname. To cancel them, you issue individual **cancel** commands that include the jobname and address space ID. This is done by the cleanup program shipped with APAR OA02375. SA for OS/390 calls the program during shutdown and pre-start of the WebSphere for z/OS base environment and J2EE servers.

The WebSphere for z/OS administration procedure

To change parameters, or install or modify applications, on a J2EE server, you use the WebSphere for z/OS Administration application running on a workstation. Follow these steps to initiate your change:

1. Create a new conversation.
2. Modify the conversation, as needed.
3. Validate the conversation.
4. Commit the conversation.
5. Activate the conversation.

Once you activate the conversation, the Administration application automatically shuts down and restarts all affected J2EE servers in order to implement their configuration changes.

Regarding automation, SA for OS/390 immediately restarts any servers it controls after the Administration application shuts them down during conversation activation. If the Administration application restart fails, the conversation is not successfully activated.

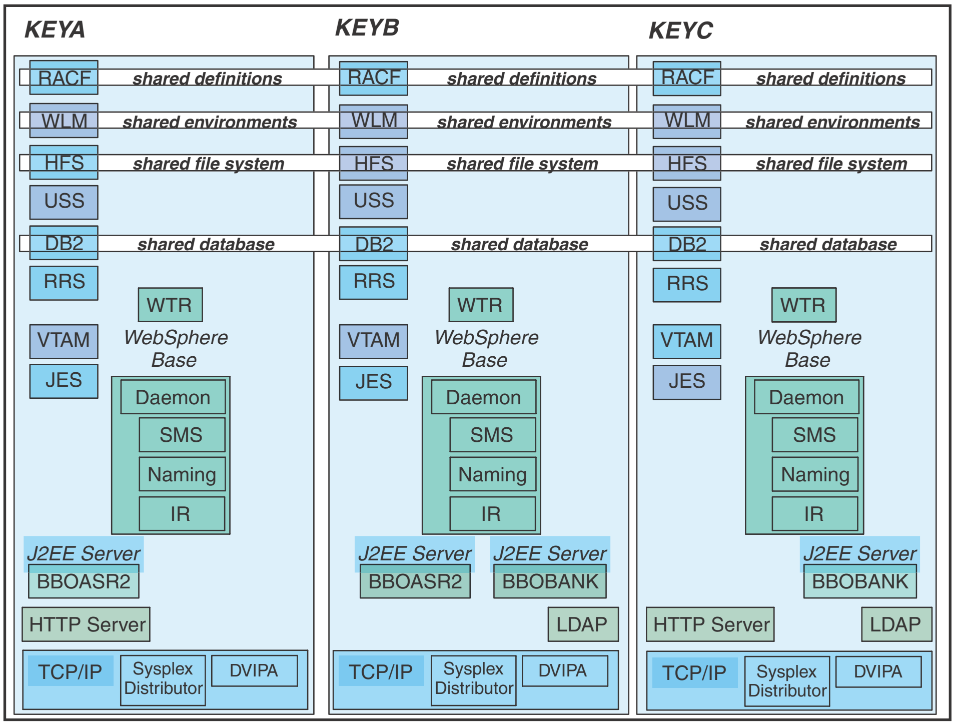


Figure 1 - Sample sysplex layout

Activating conversations manually can lead to unscheduled down times or require the presence of an administrator during scheduled maintenance periods. This conflicts with the goal of reducing down times and operational costs through system automation.

You can, however, define service periods during which activation of a previously prepared WebSphere for z/OS conversation takes place automatically:

1. Change the configuration at any time, outside the service period, using the Administration application. To do this, go through all the steps except the last conversation activation step.
2. During the service period but before activating a conversation, stop all J2EE servers through SA for OS/390 or Tivoli® Workload Scheduler (TWS).
3. Through SA for OS/390 or TWS, start a job that activates the conversation using the WebSphere for z/OS System Management Scripting API. You won't trigger a server restart through the activation because no servers are running. (Samples for jobs and execs are supplied in the download zone along with a detailed description.)
4. After you complete activation, restart all J2EE servers through SA for OS/390 or TWS.

The HTTP session concept

State information associated with a series of HTTP requests is represented as an HTTP session object. WebSphere for z/OS is responsible for managing HTTP session objects, and it offers two options: session affinity and persistent sessions.

- With session affinity, session information is held in the memory of the J2EE server that started the session. All further requests must be directed to the same server.
- With persistent sessions, session information is stored in a shared DB2 database. Each J2EE server can access it. All the objects held by a session must implement `java.io.Serializable`.

The SA for OS/390 setup implements persistent sessions. This enables you to continue a session on a J2EE server running on a different image if the server on the first image fails. It is essential for increased application availability.

What's in the sample policy database?

A policy database contains all objects managed by SA for OS/390, along with the automation rules for each object.

The main objects to consider are applications and application groups. The rules are primarily expressed through relationships. The sample PDB for WebSphere for z/OS automation defines several groups and their relationships. The main object relationships related to WebSphere for z/OS are the following: