An Automation and High Availability Solution for WebSphere Application Server for z/OS V5.1 Based on IBM Tivoli System Automation for z/OS V2.3

IBM Tivoli System Automation Development
IBM Boeblingen

October 2004

IBM Deutschland Entwicklung GmbH
Schoenaicher Strasse 220
71032 Boeblingen
Germany

The Tivoli home page on the Internet can be found at **ibm.com/Tivoli**
The IBM home page on the Internet can be found at **ibm.com**

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Contents

Figures

Chapter 1. Introduction

Running WebSphere Application Server for z/OS (WAS) in a sysplex environment increases the complexity of operating such a system. This is because WebSphere has a number of prerequisite products that it interfaces with, and also uses a number of z/OS address spaces itself to deliver high performance and high availability Web services.

When you run Internet applications on a system, its availability and its performance is exposed to the public. It is therefore essential to provide a scalable and highly available setup for operations providing Internet services.

This paper presents a system automation and high availability solution. This provides a sample of how to set up a fully automated operational environment for all WebSphere Application Server for z/OS V5.1 components and prerequisites and related products running on a z/OS environment using IBM Tivoli System Automation for z/OS (SA z/OS).
IBM supports this set-up by

- Providing this white paper as a guideline
- Supplying a sample Policy Database (PDB) *WAS51_HA through **APAR OA09374** that contains the SA z/OS definitions used to set up the environment described in the white paper.
- Enhancing the automation table by adding messages issued by WebSphere and related products during the start up and termination processing through **APAR OA09374**.

1.1 WebSphere Application Server for z/OS V5.1

WebSphere® Application Server is a comprehensive, sophisticated, Java™ 2 Enterprise Edition (J2EE) and Web services technology-based application platform specifically designed to leverage the qualities of service inherent in the z/OS operating system. V5.1 includes maintenance release V5.0.2 support for the Software Development Kit for Java Technology Edition 1.4 (SDK 1.4) client container and improvements in Web services interoperability and security, and systems management.
WebSphere Application Server for z/OS, V5.1 is designed, as was V5.0, to be equivalent from a programming model perspective with WebSphere Application Server and WebSphere Application Server Network Deployment for multiplatforms.
Additionally, V5.1 provides the following functionality

- Support for SDK 1.4 as the first step toward J2EE 1.4 compliance which can help allow enterprises to use Java technology to develop more demanding business applications with less effort and in less time
- Third-generation support for the Web services standards utilized to help transform and integrate business designs and business processes while helping to ensure business continuity through better integration with key partners, suppliers, and customers
- Extensive integration with the z/OS operating system to help leverage the high reliability, strong security, and overall robust qualities of service of the zSeries hardware and z/OS operating system

For more details on WebSphere Application Server for z/OS V5.1 please refer to http://www.ibm.com/software/webservers/appserv/zos_os390/

1.2 IBM Tivoli System Automation for z/OS V2.3

Organizations with single-processor IBM z/OS systems and IBM Parallel Sysplex clusters use IBM Tivoli System Automation for z/OS, Version 2.3, to help ease systems management, minimize costs and maximize application availability. With SA z/OS, you can automate I/O, processor and system operations. The software includes out-of-the-box automation for IMS, CICS, Tivoli Workload Scheduler, DB2, mySAP, and IBM WebSphere.
You can leverage the extensive capabilities of SA z/OS to implement a base for autonomic end-to-end automation of your e-business applications.

SA z/OS primarily deals with starting and stopping applications in accordance with their inter-relationships. These include relationships of applications to other applications, or being a component application of an application complex. SA z/OS also supports the permanent availability of an application by moving it to another system if there is an unrecoverable failure.

All applications and systems that you may want to include in automation must be defined to SA z/OS in an automation Policy Database. This database contains the objects to be managed by SA z/OS, and the rules according to that automation of these objects proceeds.

For more details on IBM Tivoli System Automation for z/OS V2.3 please refer to http://www.ibm.com/servers/eserver/zseries/software/sa/

Experiences with IBM Tivoli System Automation for z/OS V2.3 can be discussed in the SA z/OS user forum on http://groups.yahoo.com/group/SAUsers/

Chapter 2. System Setup

2.1 Sysplex

The sample sysplex consists of three systems KEYA, KEYB, and KEYC. All systems run a z/OS 1.5 and are set up symmetrically, i.e. all prerequisite products and all components of WAS are installed on each of the three systems.
Cross system sharing was implemented where applicable: RACF definitions, DB2 and MQ are shared.

Note: z/OS ARM must be disabled for those components which are managed by SA z/OS.

Note: Do not use the WAS Administration interface to start or stop WAS components. (This is done automatically by SA z/OS.)

The products used include:

- z/OS V1.5
- WebSphere Application Server for z/OS V5.1
- HTTP Server V5.3
- Tivoli NetView V5.1
- IBM Tivoli System Automation for z/OS V2.3

2.2 WebSphere Application Server for z/OS V5.1

2.2.1 Components

The following WAS components are considered:

- **Cell:** There is one network deployment cell that spans all three systems.
- **Deployment Manager:** The Deployment Manager is required for administration and deployment only. It is not necessary for running Web applications in a J2EE Server which are already deployed. One and only one Deployment Manager can run per cell. The Deployment Manager is set up for KEYA as its standard system. Through the use of VIPA it can be started on any system, on which a daemon is already running.
- **Daemon:** There is one instance needed per system per cell. The daemon is required by all other WAS components. The daemon on a system is started automatically when another WAS component is started and it is not yet up. When the daemon abends, all other running WAS components get killed automatically.
- **Node:** There are three nodes defined, one on each system. All nodes belong to the one network deployment cell.
- **Node Agent:** One Node Agent is required on each node. It is used for administration and deployment only. It is not necessary for running Web applications in a J2EE Server which are already deployed.
- **J2EE server:** There is one J2EE server defined on each node. All of them are part of the cluster. Each J2EE server comprises a Controller address space and one or more Servant address spaces. Both are required for running J2EE applications. The Controller is started externally, the Servants are started through WLM by the Controller (in this sample, a minimum of 2 and a maximum of 4 Servants are started per Controller). All servers in a cluster can be started simultaneously from the Admin Console. In this sample, only two servers are started and the third server is left as a reserve on a spare system in case of a failure on one of the original systems.

- **Cluster:** There is one cluster defined containing all J2EE servers of the network deployment cell, spanning all three systems.

In addition:

- **Component Trace Writer:** Even if this component is not strictly required, the sample provides it to avoid messages which are issued when a WAS component is started without having first started BBOWTR on the same system.
- **HTTP Server:** There is one HTTP server installed on each system. The servers can also be reached by a VIPA address.

This can be extended by:

- **Additional cells:** An additional cell means an additional WAS. It needs a duplication of the complete sample. It needs a complete infrastructure containing IP ports, file system, daemons, deployment manager…
- **Additional nodes:** Additional nodes can be added within a cell on any system to structure additional J2EE servers. A Node Agent must be defined. See “4.7 Relationships” on page 45 for necessary relationship definitions.
- **Adding Systems:** Additional systems might be provided to enhance the supporting capacity for existing applications. In this case,
    - Copy the nodes which contain the applications and adapt the cell, respectively cluster definition.
    - Within the PDB, the given Node, Node Agent, and J2EE Server definitions must only be linked to the additional system.

2.2.2 Component Naming Conventions

A component name can consist of up to 8 characters.
The systematic for the names is:

- 1st char: **B** (to have a common starting point, e.g. for D A,B*)
- 2nd char: cell number, here always **1** since we have only one cell.
- The next part identifies the object:

CELL		Network Deployment cell
NODE		node, plus further identifier:
	DM	the special node for the deployment manager
	1A	node 1 on KEYA
	1B	node 1 on KEYB
	1C	node 1 on KEYC
DMN		daemon
DMGR		deployment manager
AGT		node agent, plus further identifier:
	1A	node agent of node 1 on KEYA
	1B	node agent of node 1 on KEYB
	1C	node agent of node 1 on KEYC
SR1		J2EE server 1, plus further identifier:
	Blank	server cluster
	1A	J2EE server of node 1 on KEYA
	1B	J2EE server of node 1 on KEYB
	1C	J2EE server of node 1 on KEYC

2.2.3 Component Names

Object	Description	Short Name	Controller Proc	Servant Proc	Controller JNM	Servant JNM
Cells	ND cell	B1CELL	N/A	N/A	N/A	N/A
Nodes	Deployment Manager node	B1NODEDM	N/A	N/A	N/A	N/A
	1st (and only) node of cell 1 on KEYA	B1NODE1A	N/A	N/A	N/A	N/A
	1st (and only) node of cell 1 on KEYB	B1NODE1B	N/A	N/A	N/A	N/A
	1st (and only) node of cell 1 on KEYC	B1NODE1C	N/A	N/A	N/A	N/A
Daemons	ND Daemon	B1DMN	BBO5DMN	N/A	B1DMN	N/A
Depl. Mgrs	Deployment Manager (one for each ND cell)	B1DMGR	BBO5DCR	BBO5DSR	B1DMGR	B1DMGRS
Node Agents	Node Agent on Node A (B1NODE1A)	B1AGT1A	BBO5ACR	N/A	B1AGT1A	N/A
	Node Agent on Node B (B1NODE1B)	B1AGT1B	BBO5ACR	N/A	B1AGT1B	N/A
	Node Agent on Node C (B1NODE1C)	B1AGT1C	BBO5ACR	N/A	B1AGT1C	N/A
Clusters	Cluster 1	B1SR1	N/A	N/A	N/A	N/A
Application Servers	1st (and only) J2EE Server on node B1NODE1A	B1SR11A	BBO5ACR	BBO5ASR	B1SR11A	B1SR11AS
	1st (and only) J2EE Server on node B1NODE1B	B1SR11B	BBO5ACR	BBO5ASR	B1SR11B	B1SR11BS
	1st (and only) J2EE Server on node B1NODE1C	B1SR11C	BBO5ACR	BBO5ASR	B1SR11C	B1SR11CS

2.2.4 Component Overview

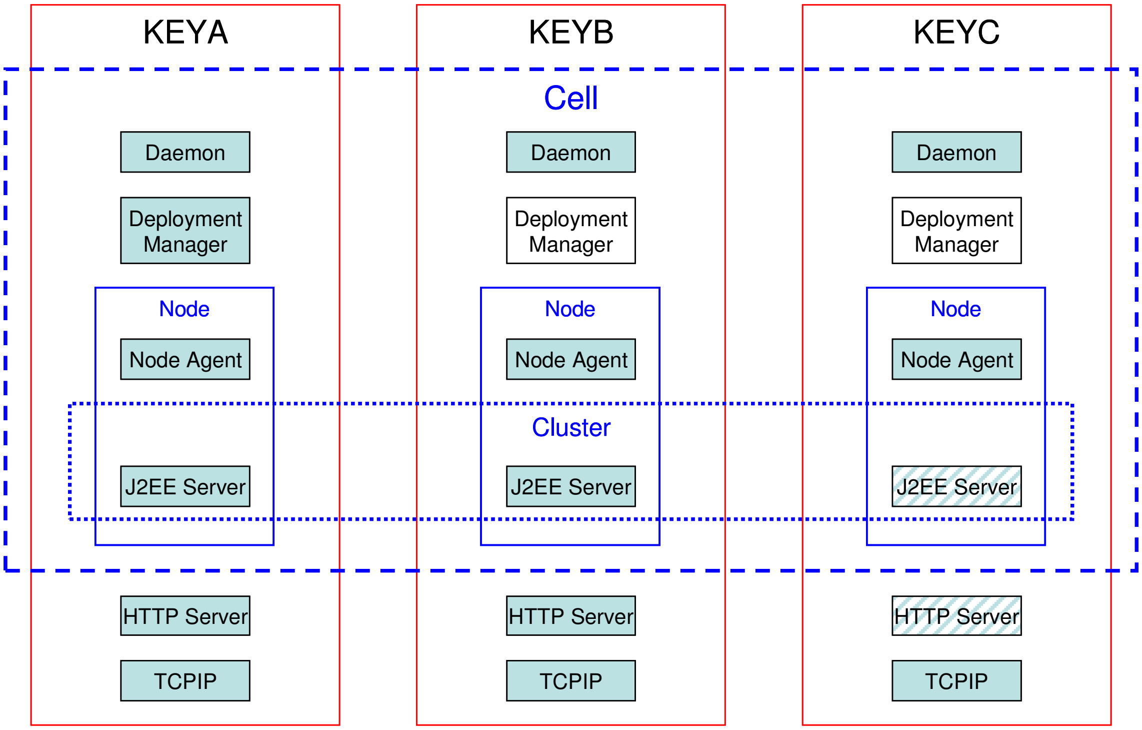

Figure 1 Component Overview

The overview shows the defined WAS components. Colouring scheme means:

- There is only one Deployment Manager available at one time. It may be moved to one of the other systems.
- An additional J2EE Server or HTTP server might be started on the third system.

2.3 HTTP Server

There are IBM HTTP Servers (with WebShpere Plugins) configured for each system. The proposed high availability solution starts two HTTP Servers and provides the third one for take over if one of the two running servers should fail.
The Sysplex Distributor sends a request to one of the available HTTP Servers which again routes this requests to the appropriate J2EE Server.

Chapter 3. Automation and High Availability Solution

Running WebSphere Application Server for z/OS (WAS) in a sysplex environment increases the complexity of operating such a system. This is because WebSphere has a number of prerequisite products that it interfaces with, and also uses a number of z/OS address spaces itself to deliver high performance and high availability Web services.

When you run Internet applications on a system, its availability and performance is exposed to the public. It is therefore essential to provide a scalable and highly available setup for operations providing Internet services.

SA z/OS can be used to **automate** an environment with WAS focusing on:

- Ease of operations support:
    - Start all prerequisites and components of WebSphere in the right order
    - Stop all components of WebSphere in the right order
    - Show status of all prerequisites and components
- Availability support reducing downtimes:
    - Monitor prerequisites and components
    - Automatically restart failing components.
- Ensuring that WebSphere Application Servers run on a predefined number of images in the sysplex
- Moving all or parts of WebSphere Application Server for OS/390 and z/OS from a failing image to another image when restart of vital component fails
- Cleaning up failed system.

The aspects of **high availability** in general include:

- Continuous operation: this covers the ability to avoid planned outages
    - Enable administrative work, and maintenance of hardware and software to be done while the application remains available to end users
    - Normally accomplished by providing multiple servers, and switching end users to an available server when one server is unavailable
- High availability: this is the ability to avoid unplanned outages by eliminating single points of failure
- The ability to minimize the effect of an unplanned outage by masking the outage from end users
- Continuous availability: this combines the characteristics of high availability and continuous operation to get as close as possible to 24x7x365.

SA z/OS enhances the capabilities that are provided by WAS especially by

- Automatic restart of failed components which is particularly needed for single system environments
- Supporting spare (backup) systems. Adding supplementary systems to the environment in case of WAS component failures prevents single point of failure situations.

Note: Workload considerations are not covered. This is handled by WLM within the J2EE Server servants.

3.1 Sequence and Grouping Considerations

General remark
The sample sysplex comprises three systems. It is set up to have two systems available for running J2EE applications while having one spare system as standby to start WAS related components when one of the other systems fails.

Base components and TCP
Base components which must be up before any WAS component is started. This includes TCP/IP and RRS. These considerations are reflected by relationship definitions in the policy.

Trace writer
The trace writer should be up on each system before any WAS component.
When the trace writer is not started, you get error messages during start up of the WAS components, but start-up continues and WAS works fine after start-up.

HTTP Server
The HTTP Server should run on two of the three systems. In case of a failure another server should be started on the spare system to prevent single point of failure.
It depends on TCP/IP being up on the specific system.

Daemon
The daemons should run on all 3 systems. It must run before any other WAS component is started.
The daemon depends on TCP/IP being up on the specific system.

Deployment Manager
The Deployment Manager must run on one and only one system. When it fails, it should be moved to one of the other systems.
The Deployment Manager depends on the daemon being up on the system where it is started. This is reflected in an appropriate relationship definition in the policy.

Node Agent
The Node Agent must run all three systems.
The Node Agent depends on the daemon being up on the system where it is started.

J2EE server
The J2EE server should be started on two of three systems. If it fails on one system it should be moved to the spare one.
The J2EE server depends on the daemon on its system. In order to be available over the web it needs the HTTP Server.

Start-up sequence

1. Start the base including TCP/IP and trace writer.
2. Start the WAS daemons and the HTTP server.
3. When the WAS daemons are running, start the WAS administration objects (deployment manager and node agents) and the operation objects (J2EE servers).

3.2 User Scenarios

When talking about starting and stopping in the following scenarios it is assumed to do this by using appropriate SA z/OS means.

3.2.1 Start and stop a complete cell

All WAS objects that belong to a specific cell are grouped into one sysplex group (named BBO_CELL). By starting or stopping this group, all contained objects are started or stopped in the order which is defined by the relationships between the single objects or object groups.

3.2.2 Take off and add in a system

For maintenance purposes it might be necessary to take off all WAS objects from a specific cell which run on a specific system. To do so, stop the daemon on that specific system. All related objects are stopped by a defined relationship and started on the spare system as defined by the availability target. Cancelling the request will make the system eligible again for running the WAS components.

3.2.3 Move the deployment manager to another system

The deployment manager is defined in a move group.
Therefore, the deployment manager can be transferred to another system by simply stopping the running instance. The deployment manager is then restarted on another system.
While the deployment manager is transferred the node agents continue to run.
Before the deployment manager can be started on another system the daemon on this specific system is started (if not already available there).

Note: The start up time of the WAS deployment manager must be considered here. Even if it is restarted immediately (what is the case with SA z/OS), it takes the remarkable start up time of about 13 minutes (measured in the sample test environment) to have the administration available again.

3.2.4 Monitor operation and administration

In order to monitor the status of WAS operation and administration, the sample policy provides two passive[1] groups, named BBO_MONO and BBO_MONA.

- BBO_MONO comprises the daemons and the J2EE servers of a specific cell. It provides a combined status of these objects representing the WAS operational capabilities.
- BBO_MONA comprises the daemons, the node agents, and the deployment manager objects of a specific cell. It provides a combined status of these objects representing the WAS administrative capabilities.

3.2.5 Provide J2EE server high availability

Clustered J2EE servers are defined in server groups. This guarantees high availability without the need of having all servers up and running. In case of a failure, another J2EE server is started on a supplementary system; if necessary, the complete WAS environment on this system is brought up.

[1] Passive groups provide only a composed status of all its objects. They cannot be used for operational purposes.

Note: The start up time of the WAS components must be considered here. A single point of failure situation is not eliminated immediately after detection. It takes the time until the additional J2EE is operational which might take several minutes.

3.2.6 Provide HTTP server high availability

The Sysplex Distributor sends the requests to one of the available HTTP Servers. The servers are grouped in a server group with an availability target of two. By this, there are two servers operational while one is left on the spare system to take over in case one of the others should fail. This eliminates the single point of failure and allows continuous availability.

3.3 Policy Overview

Note: All observed time refer to this sample sysplex and its systems. These numbers must be adapted to any other environment using the sample PDB.

See Figure 3 WAS Entries on page 17 for the application groups and application which are defined for WAS objects.

Chapter 4. IBM Tivoli System Automation for z/OS V2.3 Policy

4.1 Common Base Entries

The policy data base is built upon the common base sample which comprises the following **application groups** and ***applications***:

- **AM_PLEX** (sysplex group SERVER) is the Automation Manager group of the sysplex and contains the ***Automation Managers (AM)****.*
- **BASE_APPL** (system group BASIC) is the Base z/OS Applications group and contains ***TSO***.
- **BASE_SYS** (system group BASIC) is the Base z/OS Components group and contains ***JES2***, ***IRRDPTAB***, ***RRS***, ***RMF*** and ***RMFGAT***.
- **LOOKASID** (system group BASIC) is the LookAside components group and contains ***DLF***, ***LLA*** and ***VLF***.
- **MQ_PLEX** (sysplex group SERVER) is the WebSphere MQ group of the sysplex and contains the **MQ_SYS** group.
- **MQ_SYS** (system group BASIC) is the WebSphere MQ group and contains ***MQCHIN*** (channel initiator) and ***MQMSTR*** (queue manager).
- **NETWORK** (system group BASIC) contains ***APPC***, ***ASCH*** (APPC scheduler), ***RESOLVER***, ***TCPIP*** and ***VTAM***.
- **SGA_PLEX** (sysplex group BASIC) is the DB2 group of the sysplex and contains the **SGA_SYS** group.
- **SGA_SYS** (system group BASIC) is the DB2 group and contains ***SGA_DBM1***, ***SGA_DIST***, ***SGA_IRLM***, ***SGA_MSTR*** and ***SGA_SPAS***.
- **SYSVIEW** (system group BASIC) is the System Automation Agent Resources group and contains ***SYSVAPPL*** and ***SYSVSSI***.

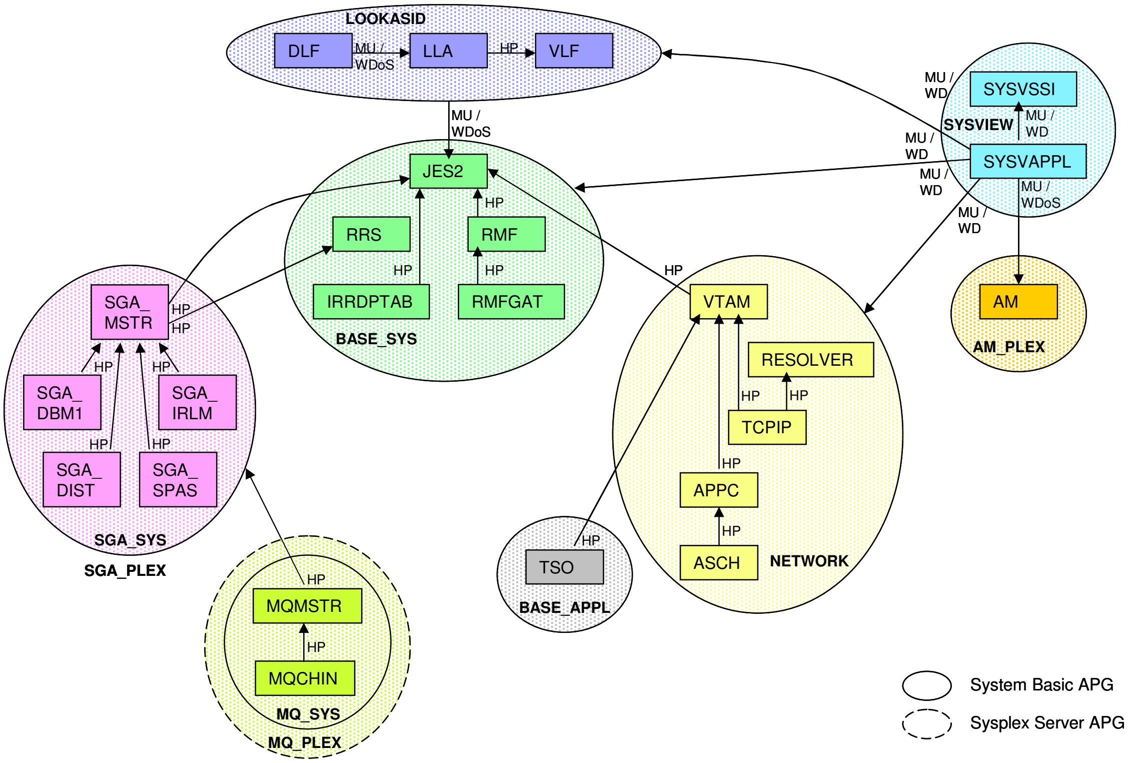

Figure 2 Common Base Entries

4.2 WebSphere Application Server Entries

The policy data base comprises the following **application groups** and ***application*** for WAS objects:

- **BASE_WAS** (system group BASIC) is the Base WAS Components group and contains the trace writer ***BBOWTR***.
- **BBO_ADMIN** (sysplex group BASIC) is the WAS Administration group and contains the node agents ***B1AGT*** and the **BBO_DMGR** group.
- **BBO_CELL** (sysplex group BASIC) is the WAS Cell group and contains the groups **BBO_ADMIN**, **BBO_DMN** and **BBO_SR1**.
- **BBO_DMGR** (sysplex group MOVE) is the WAS Deployment Manager group and contains the deployment manager ***B1DMGR***.
- **BBO_DMN** (sysplex group SERVER) is the WAS Daemon group and contains the daemons ***B1DMN***.
- **BBO_SR1** (sysplex group SERVER) is the WAS J2EE Server group and contains the application servers ***B1SR1***.
- **WEB_PLEX** (sysplex group SERVER) is the HTTP Server group and contains the HTTP servers ***WEBSRV***.

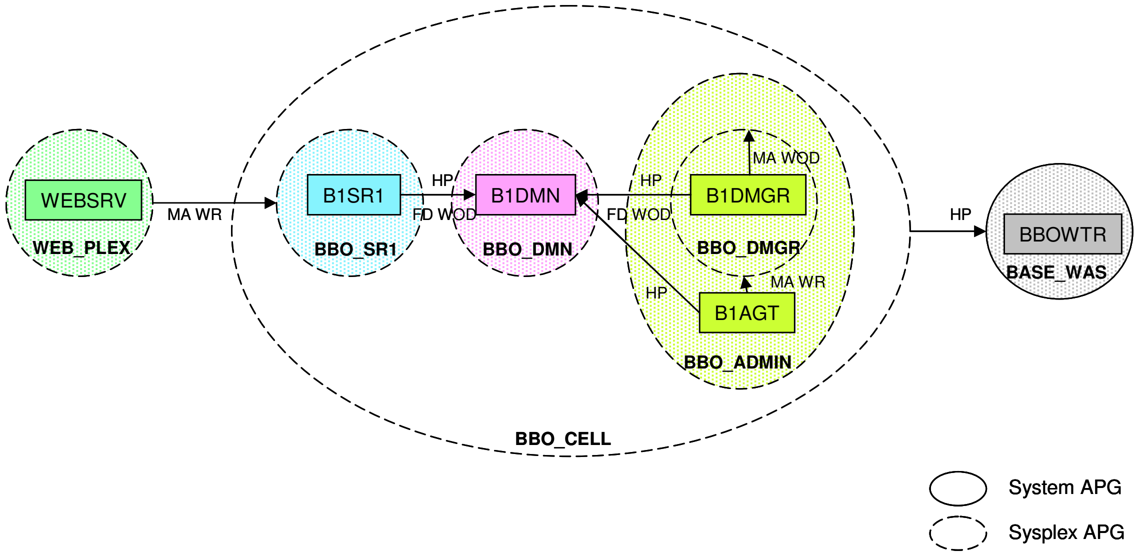

Figure 3 WAS Entries

4.3 Sysplex Groups

4.3.1 KEYAPLEX

This is the definition for the Sysplex group KEYAPLEX.

Short description	sysplex with systems KEYA, KEYB, KEYC
Sysplex Timer	NO
Desired monitoring for Sysplex	PRIMARY
Desired monitoring for ARM	NONE
Desired monitoring for CFRM	NONE
Desired monitoring for LOGR	NONE
Desired monitoring for WLM	NONE
Desired monitoring for SFM	NONE
Logical SYSPLEX Group ID	RH
Valid for ProcOps commands	NO
No of cylinders for page data set allocation	400
Group type	SYSPLEX

4.3.1.1 Linked Systems

Entry Name	Short Description
KEYA	System KEYA
KEYB	System KEYB
KEYC	System KEYC

4.3.1.2 Linked ApplicationGroups

Entry Name	Short Description
AM_PLEX	Automation Manager Group for KEYAPLEX
BBO_ADMIN	WAS V5 Administration group
BBO_CELL	WAS V5 Cell group
BBO_DMGR	WAS V5 Deployment Manager group
BBO_DMN	WAS V5 Daemon group
BBO_MONA	WAS V5 Monitor Administration group
BBO_MONO	WAS V5 Monitor Operation group
BBO_SR1	WAS V5 J2EE Server group
MQ_PLEX	MQ sysplex server group
SGA_PLEX	DB2 - SGAn sysplex server group
WEB_PLEX	IBM HTTP Server group

4.4 Systems

4.4.1 KEYA

This is the definition for the system KEYA.

Automation Symbol 2 is used for WAS purposes: Because there is one node on each system, containing one J2EE server, Automation Symbol 2 defines the suffix to the short names and to the job names of the node agents and the J2EE servers.

Short description	System KEYA
Operating system	MVS
ProcOps name	KEYA
MVS SYSNAME	KEYA
Heartbeat Interval	5
Missing Heartbeat Delay	30
Automation Symbol 0	1
Automation Symbol 1	A
Automation Symbol 2	1A
Automation Symbol Description	used for WAS V5 server name
Sys-Ops NetView Domain	IPXFG
NetView Network Name	DEIBMIPS
RMF installed	YES
OPC/ESA installed	NO
SDSF installed	YES
Initialized target system	NO
Primary JES Subsystem	JES2
System Monitor Time	00:30
Gateway Monitor Time	00:15
Monitor Option	INGPJMON
Message Table	INGMSG01
SDF Root Name	KEYA

4.4.1.1 Route codes and appropriate MCSFLAGS

Master Console information code	2
Queue message to console	1
Console ID	01

4.4.1.2 Linked ApplicationGroups

Entry Name	Short Description
BASE_APPL	Base z/OS Applications
BASE_SYS	Base z/OS Components
BASE_WAS	Base WAS V5 Components

LOOKASID	LookAside Components per system
MQ_SYS	WebSphere MQ Series system group
NETWORK	NETWORK - System Basic APG
SGA_SYS	DB2 - SGAn basic system group
SYSVIEW	SA Automation Agent Resources

4.4.1.3 Linked Monitors

Entry Name	Short Description
JES2MON	Monitor for JES2

4.4.1.4 Linked MVS Component

Entry Name	Short Description
MVS_COMPONENTS	MVS Components used in all systems

4.4.1.5 Linked System Defaults

Entry Name	Short Description
SYSTEM_DEFAULTS	System Defaults

4.4.1.6 Linked Application Defaults

Entry Name	Short Description
APPL_DEFAULTS	Application Defaults

4.4.1.7 Linked Auto Operators

Entry Name	Short Description
BASE_AUTOOPS	Default Automation Operators
FOCALPT_AUTOOPS	Focal Point Automation Operator
GATEWAY_AUTOOPS	Gateway Automation Operator
WORK_AUTOOPS	Automation Work Operators

4.4.1.8 Linked Network

Entry Name	Short Description
ALL_SYSTEMS_NETWORK	Network definitions used for all systems

4.4.1.9 Linked Status Details

Entry Name	Short Description
STATUS_DEFINITIONS	Defaults SDF status definition

4.4.1.10 Linked Notify Operators

Entry Name	Short Description
NOTIFY_OP_ALL_SYSTEM	Notify operators for all systems

4.4.2 KEYB

This is the definition for the system KEYB.

Automation Symbol 2 is used for WAS purposes: Because there is one node on each system, containing one J2EE server, Automation Symbol 2 defines the suffix to the short names and to the job names of the node agents and the J2EE servers.

Short description	System KEYB
Operating system	MVS
ProcOps name	KEYB
MVS SYSNAME	KEYB
Heartbeat Interval	5
Missing Heartbeat Delay	30
Automation Symbol 0	2
Automation Symbol 1	B
Automation Symbol 2	1B
Automation Symbol Description	used for WAS V5 server name
Sys-Ops NetView Domain	IPXFH
NetView Network Name	DEIBMIPS
RMF installed	YES
OPC/ESA installed	NO
SDSF installed	YES
Initialized target system	NO
Primary JES Subsystem	JES2
System Monitor Time	00:30
Gateway Monitor Time	00:15
Monitor Option	INGPJMON
Message Table	INGMSG01
SDF Root Name	KEYB

4.4.2.1 Route codes and appropriate MCSFLAGS

Master Console information code	2
Queue message to console	1
Console ID	01

4.4.2.2 Linked ApplicationGroups

Entry Name	**Short Description**
BASE_APPL	Base z/OS Applications
BASE_SYS	Base z/OS Components
BASE_WAS	Base WAS V5 Components
LOOKASID	LookAside Components per system

MQ_SYS	WebSphere MQ Series system group
NETWORK	NETWORK - System Basic APG
SGA_SYS	DB2 - SGAn basic system group
SYSVIEW	SA Automation Agent Resources

4.4.2.3 Linked Monitors

Entry Name	Short Description
JES2MON	Monitor for JES2

4.4.2.4 Linked MVS Component

Entry Name	Short Description
MVS_COMPONENTS	MVS Components used in all systems

4.4.2.5 Linked System Defaults

Entry Name	Short Description
SYSTEM_DEFAULTS	System Defaults

4.4.2.6 Linked Application Defaults

Entry Name	Short Description
APPL_DEFAULTS	Application Defaults

4.4.2.7 Linked Auto Operators

Entry Name	Short Description
BASE_AUTOOPS	Default Automation Operators
FOCALPT_AUTOOPS	Focal Point Automation Operator
GATEWAY_AUTOOPS	Gateway Automation Operator
WORK_AUTOOPS	Automation Work Operators

4.4.2.8 Linked Network

Entry Name	Short Description
ALL_SYSTEMS_NETWORK	Network definitions used for all systems

4.4.2.9 Linked Status Details

Entry Name	Short Description
STATUS_DEFINITIONS	Defaults SDF status definition

4.4.2.10 Linked Notify Operators

Entry Name	Short Description
NOTIFY_OP_ALL_SYSTEM	Notify operators for all systems

4.4.3 KEYC

This is the definition for the system KEYC.

Automation Symbol 2 is used for WAS purposes: Because there is one node on each system, containing one J2EE server, Automation Symbol 2 defines the suffix to the short names and to the job names of the node agents and the J2EE servers.

Short description	System KEYC
Operating system	MVS
ProcOps name	KEYC
MVS SYSNAME	KEYC
Heartbeat Interval	5
Missing Heartbeat Delay	30
Automation Symbol 0	3
Automation Symbol 1	C
Automation Symbol 2	1C
Automation Symbol Description	used for WAS V5 server name
Sys-Ops NetView Domain	IPXFI
NetView Network Name	DEIBMIPS
RMF installed	YES
OPC/ESA installed	NO
SDSF installed	YES
Initialized target system	NO
Primary JES Subsystem	JES2
System Monitor Time	00:30
Gateway Monitor Time	00:15
Monitor Option	INGPJMON
Message Table	INGMSG01
SDF Root Name	KEYC

4.4.3.1 Route codes and appropriate MCSFLAGS

Master Console information code	2
Queue message to console	1
Console ID	01

4.4.3.2 Linked ApplicationGroups

Entry Name	Short Description
BASE_APPL	Base z/OS Applications
BASE_SYS	Base z/OS Components
LOOKASID	LookAside Components per system
MQ_SYS	WebSphere MQ Series system group

NETWORK	NETWORK - System Basic APG
SGA_SYS	DB2 - SGAn basic system group
SYSVIEW	SA Automation Agent Resources

4.4.3.3 Linked Monitors

Entry Name	Short Description
JES2MON	Monitor for JES2

4.4.3.4 Linked MVS Component

Entry Name	Short Description
MVS_COMPONENTS	MVS Components used in all systems

4.4.3.5 Linked System Defaults

Entry Name	Short Description
SYSTEM_DEFAULTS	System Defaults

4.4.3.6 Linked Application Defaults

Entry Name	Short Description
APPL_DEFAULTS	Application Defaults

4.4.3.7 Linked Auto Operators

Entry Name	Short Description
BASE_AUTOOPS	Default Automation Operators
FOCALPT_AUTOOPS	Focal Point Automation Operator
GATEWAY_AUTOOPS	Gateway Automation Operator
WORK_AUTOOPS	Automation Work Operators

4.4.3.8 Linked Network

Entry Name	Short Description
ALL_SYSTEMS_NETWORK	Network definitions used for all systems

4.4.3.9 Linked Status Details

Entry Name	Short Description
STATUS_DEFINITIONS	Defaults SDF status definition

4.4.3.10 Linked Notify Operators

Entry Name	Short Description
NOTIFY_OP_ALL_SYSTEM	Notify operators for all systems

4.5 Application Groups

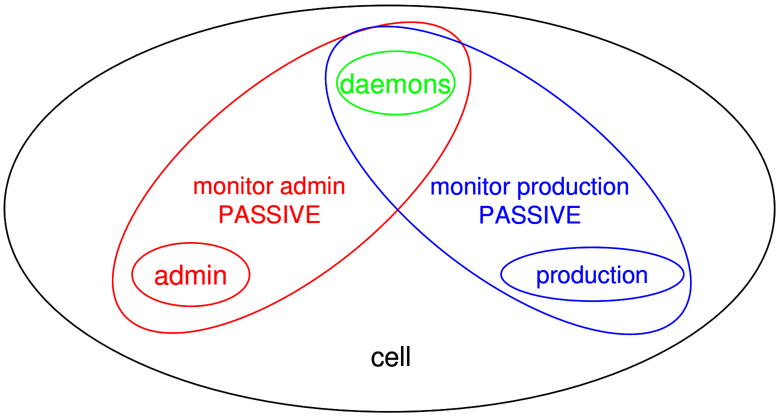

Figure 4 Application Groups

The sample PDB provides the following application groups:

- **cell** group is used to start/stop the entire WebSphere environment (related to one cell).
- **Monitor operation** group is used to monitor the run time environment.
- **Monitor admin** group is used to monitor the administration environment.
- **daemons** group is used to start/stop operability of the WebSphere environment.
- **production** group is used to start/stop operability of the WebSphere applications.
- **admin** group is used to start/stop operability of the WebSphere administration.

Base WebSphere group	BASE_WAS
Admin group	BBO_ADMIN
Cell group	BBO_CELL
Deployment manager group	BBO_DMGR
Daemons group	BBO_DMN
Monitor administration group	BBO_MONA
Monitor operation group	BBO_MONO
Production group	BBO_SR1
HTTP server group	WEB_PLEX

4.5.1 BASE_WAS

This application group contains basic WAS components such as the trace writer.

Short description	Base WAS V5 Components
Long description	Base WAS V5 Components
Application Group Type	SYSTEM
Nature	Basic
Automation Name	BASE_WAS
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.1.1 Linked Applications

Entry Name	Short Description
BBOWTR	WebSphere trace writer

4.5.1.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
BBOWTR/APL/KEYA	APL		KEYA	BBOWTR	SELECTED
BBOWTR/APL/KEYB	APL		KEYB	BBOWTR	SELECTED
BBOWTR/APL/KEYC	APL		KEYC	BBOWTR	SELECTED

4.5.1.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BASE_WAS/APG/KEYA	KEYA		
BASE_WAS/APG/KEYB	KEYB		
BASE_WAS/APG/KEYC	KEYC		

4.5.2 BBO_ADMIN

This application group contains the WAS components needed for administrative tasks.

Short description	WAS V5 Administration group
Long description	WAS V5 Administration group
Application Group Type	SYSPLEX
Nature	Basic
Automation Name	BBO_ADMIN
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.2.1 Linked Applications

Entry Name	Short Description
B1AGT	WAS V5 Node Agent

4.5.2.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
BBO_DMGR/APG	APG	KEYAPLEX		BBO_DMGR	SELECTED
B1AGT/APL/KEYA	APL	KEYAPLEX	KEYA	B1AGT	SELECTED
B1AGT/APL/KEYB	APL	KEYAPLEX	KEYB	B1AGT	SELECTED
B1AGT/APL/KEYC	APL	KEYAPLEX	KEYC	B1AGT	SELECTED

4.5.2.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_ADMIN/APG	KEYAPLEX		

4.5.3 BBO_CELL

This application group contains all WAS components of a defined cell.

Short description	WAS V5 Cell group
Long description	WAS V5 Cell group
Application Group Type	SYSPLEX
Nature	Basic
Automation Name	BBO_CELL
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.3.1 Relationships

The relationship defines that the cell can only be started if the application group BASE_WAS which contains supporting WebSphere components (in this sample: the WebSphere trace writer) is available.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
HASPARENT	BASE_WAS/APG/=					

4.5.3.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
BBO_ADMIN/APG	APG	KEYAPLEX		BBO_ADMIN	SELECTED
BBO_DMN/APG	APG	KEYAPLEX		BBO_DMN	SELECTED
BBO_SR1/APG	APG	KEYAPLEX		BBO_SR1	SELECTED

4.5.3.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_CELL/APG	KEYAPLEX		

4.5.4 BBO_DMGR

This application group contains the WAS deployment managers.

It is defined as a Move group, because there is only one instance of this component allowed within a cell.

Short description	WAS V5 Deployment Manager group
Long description	WAS V5 Deployment Manager group
Application Group Type	SYSPLEX
Nature	Move
Automation Name	BBO_DMGR
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.4.1 Linked Applications

Entry Name	Short Description
B1DMGR	WAS V5 Deployment Manager

4.5.4.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
B1DMGR/APL/KEYA	APL	KEYAPLEX	KEYA	B1DMGR	700
B1DMGR/APL/KEYB	APL	KEYAPLEX	KEYB	B1DMGR	700
B1DMGR/APL/KEYC	APL	KEYAPLEX	KEYC	B1DMGR	700

4.5.4.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_DMGR/APG	KEYAPLEX	1	

4.5.5 BBO_DMN

This application group contains the WAS daemons.

It is defined as a Server group with an availability target of *ALL, because the WAS daemon must be available on a system before any other WAS component can be started there. (Server group is preferred over basic group here, because in case not all systems are used for WAS operation, status of this group is *degraded* rather than *problem*.)

Short description	WAS V5 Daemon group
Long description	WAS V5 Daemon group
Application Group Type	SYSPLEX
Nature	Server
Automation Name	BBO_DMN
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.5.1 Linked Applications

Entry Name	Short Description
B1DMN	WAS V5 Daemon

4.5.5.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
B1DMN/APL/KEYA	APL	KEYAPLEX	KEYA	B1DMN	700
B1DMN/APL/KEYB	APL	KEYAPLEX	KEYB	B1DMN	700
B1DMN/APL/KEYC	APL	KEYAPLEX	KEYC	B1DMN	700

4.5.5.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_DMN/APG	KEYAPLEX	*ALL	*ALL

4.5.6 BBO_MONA

This application group contains all WAS components which are relevant to administrative purposes.

It is defined as a passive group, because it is used for monitoring purposes only.

Short description	WAS V5 Monitor Administration group
Long description	WAS V5 Monitor Administration group
Application Group Type	SYSPLEX
Nature	Basic
Automation Name	BBO_MONA
Behaviour	PASSIVE
Default Preference	*DEF
Automatically link	YES

4.5.6.1 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
BBO_ADMIN/APG	APG	KEYAPLEX		BBO_ADMIN	SELECTED
BBO_DMN/APG	APG	KEYAPLEX		BBO_DMN	SELECTED

4.5.6.2 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_MONA/APG	KEYAPLEX		

4.5.7 BBO_MONO

This application group contains all WAS components which are relevant to operational purposes.

It is defined as a passive group, because it is used for monitoring purposes only.

Short description	WAS V5 Monitor Operation group
Long description	WAS V5 Monitor Operation group
Application Group Type	SYSPLEX
Nature	Basic
Automation Name	BBO_MONO
Behaviour	PASSIVE
Default Preference	*DEF
Automatically link	YES

4.5.7.1 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
BBO_DMN/APG	APG	KEYAPLEX		BBO_DMN	SELECTED
BBO_SR1/APG	APG	KEYAPLEX		BBO_SR1	SELECTED

4.5.7.2 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_MONO/APG	KEYAPLEX		

4.5.8 BBO_SR1

This application group contains the WAS application servers.

It is defined as a server group with an availability target of 2, in order to use the third system in the sysplex as a spare system in case of a failure. If two J2EE servers are available then the satisfactory target is met; if only one J2EE server is available then the availability status is *degraded*.

Short description	WAS V5 J2EE Server group
Long description	WAS V5 J2EE Server group
Application Group Type	SYSPLEX
Nature	Server
Automation Name	BBO_SR1
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.8.1 Linked Applications

Entry Name	Short Description
B1SR1	WAS V5 Application Server

4.5.8.2 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
B1SR1/APL/KEYA	APL	KEYAPLEX	KEYA	B1SR1	700
B1SR1/APL/KEYB	APL	KEYAPLEX	KEYB	B1SR1	700
B1SR1/APL/KEYC	APL	KEYAPLEX	KEYC	B1SR1	700

4.5.8.3 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
BBO_SR1/APG	KEYAPLEX	2	2

4.5.9 WEB_PLEX

This application group contains the HTTP servers.

It is defined as a server group with an availability target of 2, in order to use the third system in the sysplex as a spare system in case of a failure. If two J2EE servers are available then the satisfactory target is met; if only one J2EE server is available then the availability status is *degraded*.

Short description	IBM HTTP Server group
Long description	IBM HTTP Server group
Application Group Type	SYSPLEX
Nature	Server
Automation Name	WEB_PLEX
Behaviour	ACTIVE
Default Preference	*DEF
Automatically link	YES

4.5.9.1 Relationships

The MAKEAVAILABLE relationship defines that the group is started when there is at least one J2EE server available. This ensures the accessibility of the applications which are started by the J2EE servers over the web.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
MAKEAVAILABLE	BBO_SR1/APG	start if at least 1 J2EE server running		ACTIVE	WEAK	WhenRunning

4.5.9.2 Linked Applications

Entry Name	Short Description
WEBSRV	IBM HTTP Server

4.5.9.3 Linked Resources

Resource Name	Entry Type	Sysplex name	System	Entry Name	Preference Value
WEBSRV/APL/KEYA	APL	KEYAPLEX	KEYA	WEBSRV	700
WEBSRV/APL/KEYB	APL	KEYAPLEX	KEYB	WEBSRV	700
WEBSRV/APL/KEYC	APL	KEYAPLEX	KEYC	WEBSRV	700

4.5.9.4 Resource Data

Resource Name	Sysplex/System Name	Availability Target	Satisfactory Target
WEB_PLEX/APG	KEYAPLEX	2	2

4.6 Applications

4.6.1 BBO_CLASS

BBO_CLASS represents a common policy to be inherited by the WAS V5.1 objects. It decreases the effort for defining the entries for each server individually and keeps the common definitions consistent.

Short description	WebSpere class with general definitions
Subsystem Name	BBO_CLASS
Object Type	CLASS
Application Type	STANDARD
Limit of message capturing	25

4.6.1.1 Application Messages and User Data

Message id	**Description**	**Status Msg Type**	
SHUTFORCE	Executed when force shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS C &SUBSJOB
SHUTIMMED	Executed when immediate shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS P &SUBSJOB
	2		MVS C &SUBSJOB
SHUTNORM	Executed when normal shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS P &SUBSJOB
	3		MVS C &SUBSJOB

4.6.1.2 Thresholds

Resource name	**Critical Number**	**Critical Interval**	**Frequent Number**	**Frequent Interval**	**Infrequent Number**	**Infrequent Interval**
BBO_CLASS	10	00:10	5	00:10	1	00:10

4.6.1.3 Linked Instances

Instance Name	Description
B1DMN	WAS V5 Daemon
B1AGT	WAS V5 Node Agent
B1DMGR	WAS V5 Deployment Manager
B1SR1	WAS V5 Application Server

4.6.2 BBOWTR

This is the definition of the WebSphere trace writer.

Short description	WebSphere trace writer
Subsystem Name	BBOWTR
Application Type	STANDARD
Job Name	BBOWTR

4.6.2.1 Application Messages and User Data

Message id	**Description**	**Status Msg Type**	
SHUTFORCE	Executed when force shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS TRACE CT,WTRSTOP=&SUBSJOB
	2		MVS C &SUBSJOB
SHUTIMMED	Executed when immediate shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS TRACE CT,WTRSTOP=&SUBSJOB
	2		MVS C &SUBSJOB
SHUTINIT	Executed when shutdown is initiated		
	Pass	**Automated Function**	**Command Text**
			MVS TRACE CT,OFF,COMP=SYSBBOSS
SHUTNORM	Executed when normal shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS TRACE CT,WTRSTOP=&SUBSJOB
	3		MVS C &SUBSJOB

STARTUP	Executed to initiate the startup		
	Pass	**Automated Function**	**Command Text**
			MVS TRACE CT,WTRSTART=&SUBSJOB

4.6.2.2 Relationships

The HASPARENT relationship defines that JES2 must be available before the WebSphere trace writer can be started (and that it is started in case it is not) and that the WebSphere trace writer is to be stopped before JES2 can be stopped.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
HASPARENT	JES2/APL/=					

4.6.3 B1AGT

This is the definition of the WAS node agent.

The start up time for a node agent is about 2.5 minutes and needs an appropriate start timeout definition.

Short description	WAS V5 Node Agent
Subsystem Name	B1AGT
Linked to Class	BBO_CLASS
Application Type	STANDARD
Job Name	B1AGT&AOCCLONE2.
JCL Procedure Name	BBO5ACR
Start timeout	00:03:00
Start cycles	2
Subsystem Startup Parameters	,ENV=B1CELL.B1NODE&AOCCLONE2..B1AGT&AOCCLONE2.

4.6.3.1 Application Messages and User Data

Note: There is no need to define these messages in the policy, because they are provided in the standard SA z/OS automation table.

Message id	Description	Status Msg Type
BBOO0002I	down	TERMINATED
BBOO0019I	up	UP

4.6.3.2 Relationships

The HASPARENT relationship defines that the WAS daemon must be available before the WAS node agent can be started (and that it is started in case it is not) and that the WAS node agent is to be stopped before the WAS daemon can be stopped.

The MAKEAVAIALBE relationship defines that the WAS node agent is started when there is the WAS deployment manager available.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
HASPARENT	B1DMN/APL/=					
MAKEAVAILABLE	BBO_DMGR/APG			ACTIVE	WEAK	WhenRunning

4.6.4 B1DMGR

This is the definition of the WAS deployment manager.

The start up time for a deployment manager is about 13 minutes (!) and needs an appropriate start timeout definition.

Short description	WAS V5 Deployment Manager
Subsystem Name	B1DMGR
Linked to Class	BBO_CLASS
Application Type	STANDARD
Job Name	B1DMGR
JCL Procedure Name	BBO5DCR
Start timeout	00:03:00
Start cycles	5
Subsystem Startup Parameters	,ENV=B1CELL.B1NODEDM.B1DMGR

4.6.4.1 Application Messages and User Data

Note: There is no need to define these messages in the policy, because they are provided in the standard SA z/OS automation table.

Message id	Description	Status Msg Type
BBOO0002I	down	TERMINATED
BBOO0019I	up	UP

4.6.4.2 Relationships

The FORCEDOWN relationship defines that the WAS deployment manager is forced down if the WAS daemon is not available. This is to cope with a situation we encountered in the test environment: If the WAS daemon stops then the WAS deployment manager servants are stopped properly but the WAS deployment manager controller survives in an unrecoverable state.

The HASPARENT relationship defines that the WAS daemon must be available before the WAS deployment manager can be started (and that it is started in case it is not) and that the WAS deployment manager is to be stopped before the WAS daemon can be stopped.

The MAKEAVAILABLE relationship defines that the new WAS deployment manager is only started after the hitherto has really stopped.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
FORCEDOWN	B1DMN/APL/=					When Observed Down
HASPARENT	B1DMN/APL/=					
MAKEAVAILABLE	BBO_DMGR/APG			PASSIVE	WEAK	When Observed Down

4.6.5 B1DMN

This is the definition of the WAS daemon.

The start up time for a daemon is about 5 seconds and does not need a specific start timeout definition.

Short description	WAS V5 Daemon
Subsystem Name	B1DMN
Linked to Class	BBO_CLASS
Application Type	STANDARD
Job Name	B1DMN
JCL Procedure Name	BBO5DMN
Subsystem Startup Parameters	,ENV=B1CELL.B1CELL.&SYSNAME

4.6.5.1 Application Messages and User Data

Note: There is no need to define these messages in the policy, because they are provided in the standard SA z/OS automation table.

Message id	Description	Status Msg Type
BBOO0008I	down	TERMINATED
BBOO0015I	up	UP

4.6.5.2 Relationships

The FORCEDOWN relationship defines that the WAS daemon is stopped in case TCP/IP is down. This is because the WAS daemon depends on TCP/IP and it is useless to run it while TCP/IP is unavailable.

The HASPARENT relationship defines that TCP/IP, respectively RRS must be available before the WAS daemon can be started (and that it is started in case it is not) and that the WAS daemon is to be stopped before TCP/IP, respectively RRS can be stopped.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
FORCEDOWN	RSS/APL/=					WhenObservedDown
FORCEDOWN	TCPIP/APL/=					WhenObservedDown
HASPARENT	RRS/APL/=					
HASPARENT	TCPIP/APL/=					

4.6.6 B1SR1

This is the definition of the WAS J2EE server.

The start up time for a J2EE server depends on the number of servants which are started. (2.5 minutes per controller and another 2 minutes per servant) and needs an appropriate start timeout definition.

Short description	WAS V5 Application Server
Subsystem Name	B1SR1
Linked to Class	BBO_CLASS
Application Type	STANDARD
Job Name	B1SR1&AOCCLONE2.
JCL Procedure Name	BBO5ACR
Start timeout	00:03:00
Start cycles	4
Subsystem Startup Parameters	,ENV=B1CELL.B1NODE&AOCCLONE2..B1SR1&AOCCLONE2.

4.6.6.1 Application Messages and User Data

Note: There is no need to define these messages in the policy, because they are provided in the standard SA z/OS automation table.

Message id	Description	Status Msg Type
BBOO0002I	down	TERMINATED
BBOO0019I	up	UP

4.6.6.2 Relationships

The FORCEDOWN relationship defines that the WAS J2EE server is forced down if the WAS daemon is not available. This is to cope with a situation we encountered in the test environment: If the WAS daemon stops then the WAS J2EE server servants are stopped properly but the WAS J2EE server controller survives in an unrecoverable state.

The HASPARENT relationship defines that the WAS daemon must be available before the WAS J2EE server can be started (and that it is started in case it is not) and that the WAS J2EE server is to be stopped before the WAS daemon can be stopped.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
FORCEDOWN	B1DMN/APL/=					WhenObservedDown
HASPARENT	B1DMN/APL/=					

4.6.7 WEBSRV

This is the definition of the IBM HTTP Server.

Short description	IBM HTTP Server
Long description	IBM HTTP Server
Subsystem Name	WEBSRV
Application Type	STANDARD
Job Name	WEBSRV
JCL Procedure Name	WEBSRV5

4.6.7.1 Application Messages and User Data

Message id	Description	Status Msg Type	
SHUTFORCE	Executed when force shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS C &SUBSJOB
SHUTIMMED	Executed when immediate shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS C &SUBSJOB
SHUTNORM	Executed when normal shutdown is invoked		
	Pass	**Automated Function**	**Command Text**
	1		MVS P &SUBSJOB
	2		MVS C &SUBSJOB

4.6.7.2 Relationships

The FORCEDOWN relationship defines that the HTTP server is stopped in case TCP/IP is down. This is because the HTTP server depends on TCP/IP and it is useless to run it while TCP/IP is unavailable.

The HASPARENT relationship defines that TCP/IP must be available before the HTTP server can be started (and that it is started in case it is not) and that the HTTP server is to be stopped before TCP/IP can be stopped.

Relationship Type	Supporting Resource	Description	Sequence Number	Automation	Chaining	Satisfy condition
FORCEDOWN	TCPIP/APL/=	stop HTTP server if TCPIP fails				WhenObservedDown
HASPARENT	TCPIP/APL/=	HTTP server is dependent on TCPIP				

4.7 Relationships

4.7.1 Common Base and WebSphere Application Server for z/OS V5.1 Objects

The WAS objects are anchored to the common base objects by defining relationships to TCP/IP, to JES2 and to RRS.

- TCP/IP must be up before starting WAS. Shutdown WAS before shutting down TCP/IP. So, iff a daemon or an HTTP server is started and TCP/IP is not yet active then TCP/IP is started first. If TCP/IP is observed down then also the daemons and the HTTP servers are stopped.
- JES2 is a required subsystem for the trace writer. If a trace writer is started and JES2 is not yet active then JES2 is started first.
- RRS is a required subsystem, so WAS will not run without it. If RRS goes down, WAS will get fatal errors. So, if a daemon is started and RRS is not yet active then RRS is started first. If RRS is observed down then also the daemons are stopped.

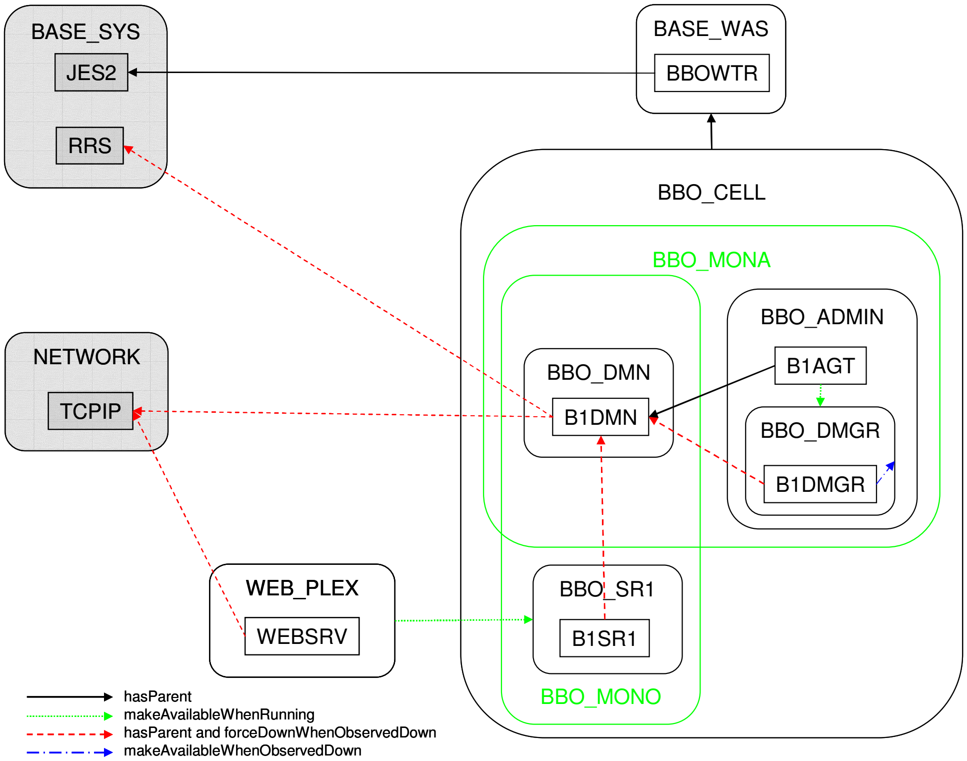

Figure 5 Common Base and WAS Objects

4.7.2 WebSphere Application Server for z/OS V5.1 Objects

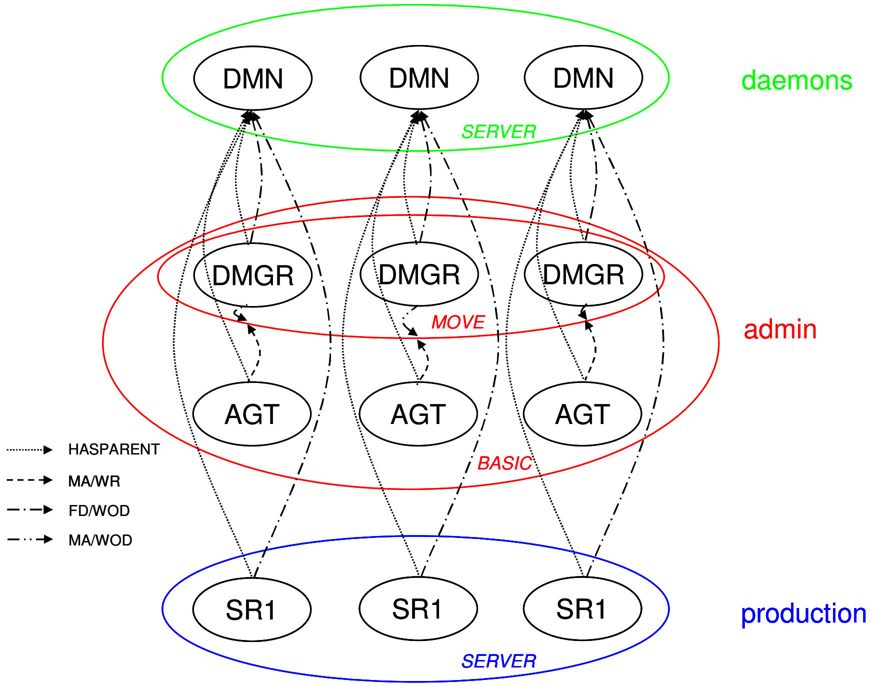

Figure 6 WAS Objects

- The daemon is necessary on each system for the WebSphere objects to run. Therefore there is a (local) HASPARENT relationship defined between these objects and the daemon.
- In our test environment we encountered the situation that the controllers of the deployment manager and J2EE servers are left in an unrecoverable state if the daemon gets down. To cope with this situation a forceDownWhenObservedDown relationship is defined to cancel the appropriate servers so that they can be properly restarted when the daemon is up again.
- The deployment manager provides the administration interface of the WebSphere cell. Full functional capability is given when the deployment manager and all node agents are up and running. Therefore there is a makeAvailableWhenRunning relationship defined between the node agents and the deployment manager group. (Note: There is only one deployment manager per cell running.)
- To make sure that the stopped deployment manager is really down before the new deployment manager is started a makeAvailableWhenObservedDown relationship is defined between the single deployment managers and their group.

Chapter 5. Message Automation

The following messages are provided in the automation table with **APAR OA09374**:

- **BBOO0002I** WAS server down message
  (used by node agent, deployment manager and J2EE server)
- **BBOO0008I** WAS daemon down message
  (used by daemon)
- **BBOO0015I** WAS daemon up message
  (used by daemon)
- **BBOO0019I** WAS server up message
  (used by node agent, deployment manager and J2EE server)