

WebSphere Profiles

Version 1.0



DigiTalk Systems

<https://digitalksystems.com/>

Reach us at: digitalk.fmw@gmail.com

DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw

WebLogic Server Architecture: <https://youtu.be/gNqeIfLjUqw>

Table of Contents

Cell Profile.....
Management Profile.....
Application Server Profile.....
Custom Profile.....
Secure Proxy Profile.....
Comparison of Profile Types.....
Additional Notes.....

DigiTalk Udemy Courses and Coupon Code

Docker Administration

<https://www.udemy.com/course/mastering-docker-administration-from-fundamentals-to-expert/?referralCode=12387A8B195F7B90381D>

Coupon Code: 4400C856A93FC2F851F3

Kubernetes Administration

<https://www.udemy.com/course/mastering-kubernetes-administration-from-intro-to-advanced/?referralCode=7E0250BEC21F46192834>

Coupon Code: 44C67B0AA69362E8BDE1

WebSphere Administration

<https://www.udemy.com/course/mastering-ibm-websphere-9x-administration/?referralCode=F372B3EA78D7BEA92416>

Coupon Code: 9C42313FDEFC9DCE4A21

Mastering SSL (TLS), Keys and Certificates Management

<https://www.udemy.com/course/mastering-ssl-tls-keys-and-certificates-management/?referralCode=DB9F1429CB01921969F6>

Coupon Code: 5B702665F8F721B6F315

SOA Suite Administration

<https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?referralCode=2BCF3895067AA7BC22E0>

Coupon Code: BF1D2646A214C279C7B1

JBoss 8 Administration

<https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?referralCode=D095CE8B1EE9D065C822>

Coupon Code: 87936B19622155F1DC68

OHS Administration

<https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?referralCode=165F6A3A8B662CEA1489>

Coupon Code: 33542ABA75949ED745F8

WebLogic Server Administration

<https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?referralCode=340FBB94DBFEE1A8CC09>

Coupon Code: BB5CB01A2CA9498DBFED

Mastering Oracle Internet Direct (OID) Administration

<https://www.udemy.com/course/mastering-oracle-internet-direct-oid-administration/?referralCode=906E9AC0059A29A3CD0D>

Coupon Code: 8D43AFACC93B009932E1

For any query or concerns please write us on digitalk.fmw@gmail.com

IBM WebSphere Application Server (WAS) uses profiles to define and manage runtime environments for applications. Each profile type serves a specific purpose, enabling flexibility in configuring and managing application servers, nodes, and cells. Below are detailed explanations of the different profile types: **Cell, Management, Application Server, Custom, and Secure Proxy**.

1. Cell Profile

A **Cell Profile** is a composite profile that combines a **Deployment Manager Profile** and a **Federated Application Server Profile** into a single configuration. It is primarily used in **WebSphere Application Server Network Deployment (WAS ND)** environments to create a distributed and managed setup.

Key Characteristics:

- **Structure:**
 - Consists of two profiles created in one process:
 - **Deployment Manager Profile:** Hosts the Deployment Manager, which manages the cell and its components via the administrative console.
 - **Federated Application Server Profile:** Includes a standalone application server that is automatically federated to the cell.
 - The federation process links the application server to the Deployment Manager, making it a managed node.
- **Purpose:**
 - Used to create a fast development or test environment where a distributed setup is needed.
 - Simplifies the creation of a cell with both a Deployment Manager and an application server in a single step.
- **Components:**
 - A **cell** (logical grouping of nodes).
 - A **Deployment Manager** (centralized management server).
 - A **node** with a federated application server (e.g., server1).
 - Optional default applications (e.g., DefaultApplication) and sample applications for testing.
- **Configuration:**
 - Stores cell-level configuration in the Deployment Manager profile.
 - Maintains a master copy of all applications and node configurations.
- **Limitations:**
 - Created on a single host; cannot span multiple physical servers during creation.
 - Requires WAS ND for full functionality (not available in Base or Express editions).

Use Case:

- Ideal for development or test environments where a quick setup of a managed, distributed environment is needed.
- Useful for administrators who want to simulate a production-like cell structure on a single machine.

Creation:

- Can be created using the **Profile Management Tool (PMT)** or the `manageprofiles.sh` command with the cell template.
- Example command:

```
./manageprofiles.sh -create -profileName Cell01 -templatePath ../profileTemplates/cell
```

- Result: Creates a Deployment Manager profile (e.g., Dmgr01) and an Application Server profile (e.g., AppSrv01) federated to the cell.

Verification:

- Check the INSTCONFSUCCESS message in the log file:
<WAS_HOME>/logs/manageprofiles/<ProfileName>_create.log.
- Verify the existence of profile directories (e.g., /opt/IBM/WebSphere/AppServer/profiles/Dmgr01 and /opt/IBM/WebSphere/AppServer/profiles/AppSrv01).
- Start the Deployment Manager and application server to ensure proper federation.

2. Management Profile

A **Management Profile** defines a runtime environment for administrative components, such as the **Deployment Manager**, **Administrative Agent**, or **Job Manager**. It is used to manage other application servers or nodes.

Key Characteristics:

- **Types of Management Profiles:**
 - **Deployment Manager Profile:**
 - Hosts the **Deployment Manager**, which manages a cell of application servers and nodes.
 - Provides the administrative console for configuring and monitoring the cell.
 - Centralizes configuration and application management for all federated nodes.
 - **Administrative Agent Profile:**
 - Manages multiple standalone application servers from a single interface.
 - Used in non-distributed environments (e.g., WAS Base or Express) to centralize administration of unfederated servers.
 - **Job Manager Profile:**
 - Coordinates administrative tasks across multiple Deployment Managers and standalone application servers.
 - Useful for managing large-scale environments with multiple cells or standalone servers.
- **Purpose:**
 - Provides centralized administration for distributed or standalone environments.
 - Enables scalability by allowing management of multiple nodes or servers.
- **Components:**
 - Administrative console (for Deployment Manager and Administrative Agent).
 - Configuration files for managing nodes, servers, or jobs.
 - No application runtime (does not host user applications).
- **Configuration:**
 - Stores configuration data specific to the management role (e.g., cell configuration for Deployment Manager).
 - Communicates with node agents (in WAS ND) or directly with servers (in Administrative Agent setups).

Use Case:

- **Deployment Manager:** Managing a production cell with multiple nodes and application servers.
- **Administrative Agent:** Simplifying administration of multiple standalone application servers in a single-server environment.
- **Job Manager:** Coordinating tasks across multiple cells or servers in large enterprise setups.

Creation:

- Created using PMT or manageprofiles.sh with the management template and the appropriate -serverType parameter:
 - For Deployment Manager: -serverType DEPLOYMENT_MANAGER
 - For Administrative Agent: -serverType ADMIN_AGENT
 - For Job Manager: -serverType JOB_MANAGER

- Example command (Deployment Manager):

```
./manageprofiles.sh -create -profileName Dmgr01 -templatePath ../profileTemplates/management -  
serverType DEPLOYMENT_MANAGER
```

Verification:

- Check the INSTCONFSUCCESS message in the log file.
- Start the management server (e.g., ./startManager.sh) and access the administrative console.

3. Application Server Profile

An **Application Server Profile** defines a standalone application server that runs Java EE applications. It is the most common profile type for deploying applications.

Key Characteristics:

- **Structure:**
 - Creates a single, standalone application server (e.g., server1) running in its own JVM.
 - Includes an independent runtime environment with its own configuration files and directories.
- **Purpose:**
 - Hosts and executes Java EE applications (e.g., servlets, EJBs, JSPs).
 - Can operate in **unmanaged mode** (standalone) or **managed mode** (federated to a Deployment Manager).
- **Components:**
 - Application server process (e.g., server1).
 - Administrative console (optional) for managing the server.
 - Default applications (optional, e.g., DefaultApplication) for testing.
 - Configuration files for resources, security, and logging.
- **Configuration:**
 - Fully self-contained, with its own Java Runtime Environment (JRE) and libraries.
 - Can be federated to a Deployment Manager to become part of a cell, at which point it is managed centrally.
- **Scalability:**
 - In standalone mode, operates independently.
 - In federated mode, supports clustering and load balancing in WAS ND.

Use Case:

- Deploying and testing applications in a single-server environment.
- Running production applications in a standalone setup (WAS Base/Express) or as part of a managed cell (WAS ND).
- Development environments where quick application deployment is needed.

Creation:

- Created using PMT or manageprofiles.sh with the default template.
- Example command:

```
./manageprofiles.sh -create -profileName AppSrv01 -templatePath ../profileTemplates/default
```

Verification:

- Check the INSTCONFSUCCESS message in the log file.
- Start the application server (e.g., ./startServer.sh server1) and deploy a test application.
- Access the administrative console or verify application deployment.

4. Custom Profile

A **Custom Profile** defines an **empty node** without any application server instances. It is designed to be federated to a Deployment Manager for further customization.

Key Characteristics:

- **Structure:**
 - Creates an empty node with no default application server (e.g., no server1 process).
 - Contains minimal configuration, making it a lightweight starting point.
- **Purpose:**
 - Provides a flexible base for creating tailored environments.
 - Allows administrators to define application servers, clusters, or other resources after federation.
- **Components:**
 - Node configuration files.
 - No default applications or servers.
 - Requires federation to a Deployment Manager to become operational.
- **Configuration:**
 - After federation, the Deployment Manager's administrative console is used to create application servers, clusters, or other resources on the node.
 - Highly customizable for specific needs (e.g., network settings, security, resource allocation).
- **Federation:**
 - Must be federated to a Deployment Manager using the addNode.sh command or during profile creation.
 - Once federated, it becomes a managed node under the cell.

Use Case:

- Creating nodes in a distributed environment where specific server configurations are needed.
- Setting up complex deployments with multiple application servers or clusters.
- Environments requiring fine-grained control over node and server configurations.

Creation:

- Created using PMT or manageprofiles.sh with the managed template.
- Example command:

```
./manageprofiles.sh -create -profileName Custom01 -templatePath ../profileTemplates/managed
```

- Federation command (post-creation):

```
./addNode.sh <dmgr_host> <soap_port>
```

Verification:

- Check the INSTCONFSUCCESS message in the log file.
- Verify the profile directory (e.g., /opt/IBM/WebSphere/AppServer/profiles/Custom01).
- After federation, use the Deployment Manager's console to create servers or clusters.

5. Secure Proxy Profile

A **Secure Proxy Profile** defines a proxy server that handles requests from the internet and forwards them to application servers, typically residing in the **Demilitarized Zone (DMZ)**.

Key Characteristics:

- **Structure:**
 - Creates a **Secure Proxy Server** (configuration-only or runtime).
 - Operates as an intermediary between external clients and internal application servers.
- **Purpose:**
 - Enhances security by isolating application servers from direct internet access.
 - Handles HTTP/HTTPS requests and forwards them to appropriate application servers.
- **Components:**
 - Proxy server process.
 - Configuration for routing rules, security settings (e.g., SSL), and load balancing.
 - No application hosting (purely a proxy).
- **Configuration:**
 - Supports SSL for secure communication.
 - Configured via the Integrated Solutions Console or exported for use in the DMZ.
 - Can be integrated with a Deployment Manager for centralized management.
- **Types:**
 - **Configuration-only Profile:** Used to configure the proxy settings, which are then exported to a runtime proxy server in the DMZ.
 - **Runtime Profile:** Runs the proxy server in the DMZ to handle actual traffic.

Use Case:

- Securing web applications by placing a proxy in the DMZ to handle external traffic.
- Load balancing requests across multiple application servers.
- Environments requiring strict security and isolation of application servers.

Creation:

- Created using PMT or manageprofiles.sh with the secureproxy template.
- Example command:

```
./manageprofiles.sh -create -profileName SecureProxy01 -templatePath  
.../profileTemplates/secureproxy
```

Verification:

- Check the INSTCONFSUCCESS message in the log file.
- Start the proxy server and test request forwarding to application servers.
- Verify SSL configuration and routing rules in the administrative console.

Comparison of Profile Types

Profile Type	Purpose	Key Components	Use Case	WAS Edition
Cell	Combines Deployment Manager and federated application server	Deployment Manager, federated node, server1	Dev/test distributed environments	ND
Management	Administrative management (DMgr, Admin Agent, Job Manager)	Admin console, configuration files	Centralized administration	ND/Base/Express
Application Server	Runs Java EE applications	Application server (e.g., server1), console	Application deployment (standalone/managed)	All
Custom	Empty node for federation and customization	Node configuration, no default server	Complex, tailored deployments	ND
Secure Proxy	Proxy for secure request forwarding	Proxy server, routing rules, SSL	DMZ security, load balancing	ND

Additional Notes

- **Profile Creation Tools:**
 - **Profile Management Tool (PMT):** A GUI tool for creating and managing profiles, located at <WAS_HOME>/bin/ProfileManagement/pmt.sh.
 - **manageprofiles Command:** A command-line tool for scripted or silent profile creation, located at <WAS_HOME>/bin/manageprofiles.sh.
- **Profile Directory:**
 - Each profile is stored in a separate directory under <WAS_HOME>/profiles/<ProfileName>.
 - Contains configuration files, logs, and runtime data specific to the profile.
- **Federation:**
 - Application Server and Custom profiles can be federated to a Deployment Manager to become part of a cell.
 - Federation is performed using the addNode.sh command or during profile creation.
- **Security:**
 - Profiles support administrative security (e.g., enabling with -enableAdminSecurity during creation).
 - Secure Proxy profiles emphasize SSL and secure routing for external traffic.
- **WAS Editions:**
 - **Base/Express:** Support standalone Application Server and Management (Administrative Agent) profiles.
 - **Network Deployment (ND):** Supports all profile types, including Cell, Custom, and Secure Proxy, for distributed environments.

Disclaimer and Consent

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with IBM WebSphere. The Company wishes to emphasize that this document is not affiliated with IBM in any way, and the content contained herein is based solely on publicly available product documentation provided by IBM.

While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with IBM's official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitaltalk.fmw@gmail.com.

We would like to make it clear that the code snippets and examples used in this document are sourced from publicly available IBM demo applications, provided by IBM for educational and learning purposes. These materials are subject to IBM's copyright and licensing terms.

We would like to emphasize that our charges are solely for the efforts and resources invested in preparing the lab documents and explanations to enhance the learning experience. DigiTalk is not charging for the usage of IBM's copyrighted material.

By using or accessing these documents, you acknowledge and agree that the IBM code snippets and examples are the intellectual property of IBM. We do not claim ownership of the code, and any trademarks or copyrights associated with IBM's remain the property of IBM.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by IBM.

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitaltalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.