

SALT Deployment File Reference



The following sections provide SALT Deployment File reference information:

- [Overview](#)
- [SALT SALTDEPLOY Format](#)
- [XML Schema](#)

Overview

The SALT Deployment File (`SALTDEPLOY`) is an XML-based file used to define SALT `GWWS` server deployment information on a per Tuxedo machine basis. `SALTDEPLOY` does the following:

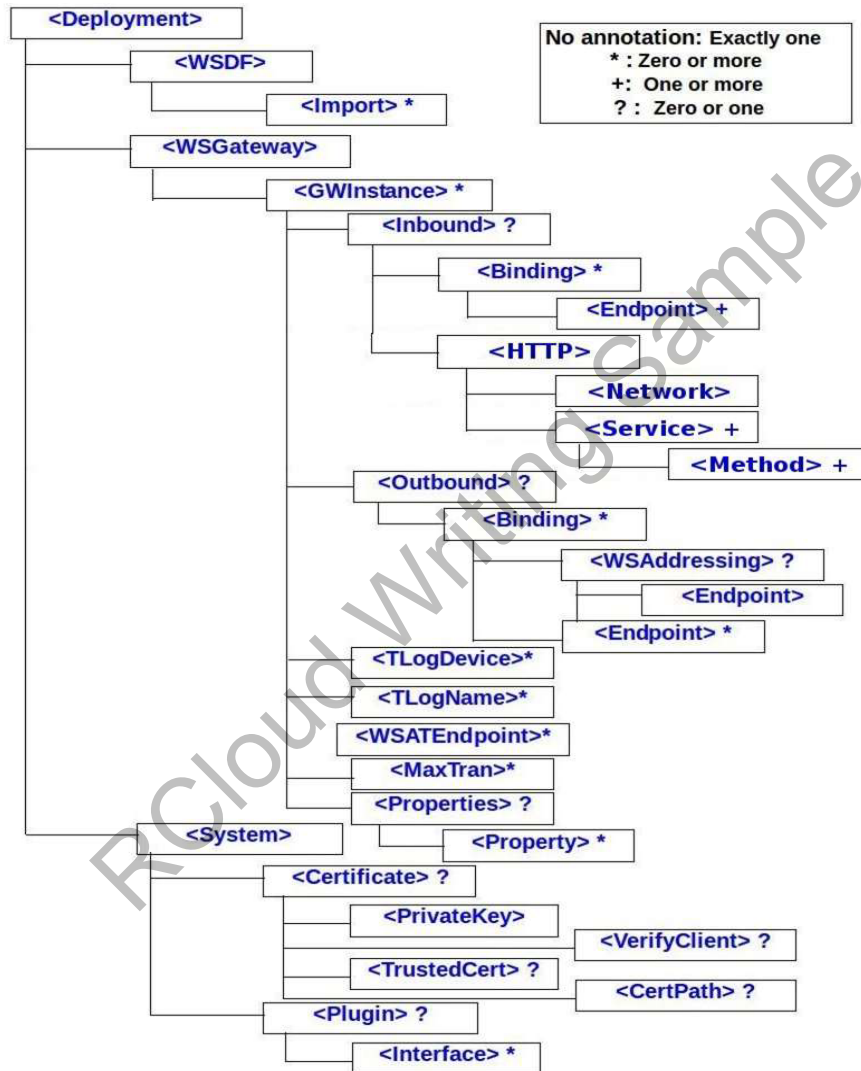
- lists all necessary Web Service Definition Files (`WSDF`)
- specifies how many `GWWS` servers are deployed on a Tuxedo machine
- associates inbound and outbound Web Service access endpoints for each `GWWS` server.

`SALTDEPLOY` also provides a system section to configure global resources (for example certificates, plug-in load libraries, and so on).

SALT SALTDEPLOY Format

[Figure B-1](#) shows a graphical representation of the SALT `SALTDEPLOY` file format.

Figure B-1 SALT Deployment File Format



XML Schema

An XML Schema is associated with a SALT SALTDEPLOY file. The XML Schema file that describes the SALT SALTDEPLOY file format is located in the following directory:

\$TUXDIR/udataobj/salt/saltdep.xsd.

[SALT SALTDEPLOY Example](#)

[SALT SALTDEPLOY Element Descriptions](#)

[<Deployment>](#)

SALT SALTDEPLOY Example

[Listing B-1](#) shows a sample SALT SALTDEPLOY File.

Listing B-1 SALT SALTDEPLOY File Example

```
<Deployment xmlns="http://www.bea.com/Tuxedo/SALTDEPLOY/2007">
  <WSDF>
    <Import location="/home/myapp/bankapp.wsdf" />
    <Import location="/home/myapp/amazon.wsdf" />
  </WSDF>
  <WSGateway>
    <GWInstance id="GW1">
      <Inbound>
        <Binding ref="bankapp:bankapp_binding">
          <Endpoint use="http1"/>
          <Endpoint use="https1" />
        </Binding>
      </Inbound>
      <Outbound>
        <Binding ref="amazon:default_binding"/>
      </Outbound>
    </GWInstance>
  </WSGateway>
</Deployment>
```

```

    <Properties>
      <Property
        name="socksAddrList"
        value="proxy.server.com,10.123.10.10:1080"/>
    </Properties>
  </GWInstance>
</WSGateway>
<System>
  <Certificate>
    <PrivateKey>/home/user/cert.pem</PrivateKey>
  </Certificate>
  <Plugin>
    <Interface library="/home/user/mydatahandler.so" />
  </Plugin>
</System>
</Deployment>

```

SALT SALTDEPLOY Element Descriptions

SALTDEPLOY format elements and their attributes are listed and described in the following section.

<Deployment>

The SALTDEPLOY file root element.

There is no attribute for this element.

Three sections must be defined within the <Deployment> element:

- <WSDF> elements
- <WSGateway> element

- `<System>` element.

There can be only one `<Deployment>` element defined in a `SALTDEPLOY` file.

`<WSDF>`

Top element that encapsulates all imported WSDF files.

There is no attribute for this element.

`<Import>`

Specifies the WSDF to be imported in the `SALTDEPLOY` file. Multiple WSDF can be imported at the same time. Each WSDF file can only be imported once. Multiple WSDF with the same WSDF name cannot be imported in the same `SALTDEPLOY` file.

Table B-1 `<Import>` Attributes

Attribute	Description	Required
<code>location</code>	Specifies the WSDF local file path.	Yes

`<WSGateway>`

Top element that encapsulates all GWWS instance definitions.

There is no attribute for this element.

`<GWInstance>`

Specifies a single GWWS instance.

Table B-2 `<GWInstance>` Attributes

Attribute	Description	Required
<code>id</code>	Specifies the GWWS identifier. This attribute value may contain a maximum of 12 characters (excluding the terminating NULL character). The identifier value must be unique within the <code>SALTDEPLOY</code> file.	Yes

<Inbound>

Specifies inbound WSBinding objects for the GWWS server. Each inbound WSBinding object is specified using the <Binding> sub element.

There is no attribute for this element.

<HTTP>

Specifies a list of services accessible in REST mode. All Oracle Tuxedo service names specified in this element are callable using HTTP or REST mechanisms. Any URL specifying a service not present on this list results in a 404 error for the caller. Any URL specifying a service present on this list (for which an Oracle Tuxedo service is not advertised), results in a 451 error for the caller.

There is no attribute for this element.

<Network>

This element contains two attributes specifying http or https (for SSL), HTTP/REST listening endpoints. Only one <Network> element is allowed per GWWS instance. The http and https elements are optional, but at least one must be specified. The http and https attributes are constructed as follows:

<host>: The name or IP address of the HTTP/REST listening endpoint.

<port>: The port value of the HTTP/REST listening endpoint.

All HTTP/REST requests are performed by the same <host>:<port> combination (i.e., it is not possible to use more than one such combination per gateway, per protocol (http and https)).

Table B-3 <Network> Attributes

Attribute	Description	Required
http	HTTP host and port listening endpoint for REST requests. Format is a string containing a <host>:<port> pair corresponding to: <host> = name or IP address of the HTTP/REST listening endpoint. <port> = port value of the HTTP/REST listening endpoint.	No*
https	SSL HTTP host and port listening endpoint specification. Same format as http attribute.	No*

* While not required, the `<Network>` element must contain either an `http` or `https` attribute.

<Service>

Specifies a single service callable using HTTP/REST mechanisms. The actual Oracle Tuxedo service called is further qualified by an HTTP method as specified using the `<Method>` element.

Table B-4 <Service> Attributes

Attribute	Description	Required
name	Name of Tuxedo service being advertised. Note: This is not the actual Oracle Tuxedo service configured using the <code><Method></code> element.	Yes
method	HTTP method to be employed in the corresponding call: 'GET', 'DELETE', 'POST' or 'PUT'.	
outputbuffer	Tuxedo buffer type/optionally subtype used for response message conversion. Values will be the same as all existing Tuxedo buffer types. For VIEW/VIEW32 buffer types, the notion of subtype will be conveyed by using the notation: {VIEW VIEW32}/<Subtype>, for example: 'VIEW32/customer'.	
address	URL of HTTP service being accessed. Supported schemes are 'http' and 'https'.	
content-type	'JSON' for JSON mapping, and 'XML' for XML mapping. This is optional and only valid for VIEW/VIEW32/X_C_TYPE/X_COMMON and FML/FML32 type buffers.	

<Method>

Specifies the HTTP method mapping to Oracle Tuxedo services. This is designed to model CRUD methods (Create, Read, Update, Delete).