



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Administrator Extension Developer's Guide

Authoring Solution Definitions

12/19/2025

Authoring Solution Definitions

Contents

- **1 Authoring Solution Definitions**
 - **1.1 Authoring an SPD File**
 - **1.2 Complete SPD samples**
 - **1.3 References**

A solution definition file (also called a solution package definition, or SPD) is an XML file that describes what is being deployed and how the deployment is to be executed, as well as any necessary pre- and post-installation procedures.

This page helps you to author your first SPD file to use with Genesys Administrator Extension.

Authoring an SPD File

Create a new SPD file.

Launch a text or XML editor, open a new document, and add the following header lines: **[+] Click here to reveal code**

```
<?xml version="1.0" encoding="utf-8"?>
<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
    xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg"
    xmlns:fileio="http://genesys.com/gax/asd/fileio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
    http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd">
</solutionDefinition>
```

The first line is the header information that is required for all XML documents. The second line indicates the schema version of the XML document, and will be used to compare against the running GAX instance. Any major version that contains a later version of the schema will be rejected by GAX.

Important

See the [General Elements](#) reference page for more information about the tags that are used in an SPD file.

Add information to personalize the SPD file.

Personalize the SPD file by using the following code as an example: **[+] Click here to reveal code**

```
<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
    xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg"
    xmlns:fileio="http://genesys.com/gax/asd/fileio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
    http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd">
    <deploymentkey>HostedProviderEdition8.1.3</deploymentkey>
    <name>DemoTest</name>
    <version>1.1.1</version>
    <description>This is the demo service package. This service package is
```

```
intended for testing the GAX automated deployment feature. It will perform one pre-
installation task, one post-installation task, and an installation of the SIP Server
application</description>
    <summary>This is the service package definition file to deploy an Demo
solution</summary>
</solutionDefinition>
```

Important

See the [General Elements](#) reference page for more information about the tags that are used in an SPD file.

Notes

- The <deploymentkey> tag is the key string that is used to match different versions of the same solution.
- The <name> tag is the name that is displayed when the XML file is imported into GAX. The name and the version uniquely identify an SPD file in GAX.
- The <version> tag indicates the version number of the SPD file.
- The <description> tag is the description text that is displayed in the user interface.
- The <summary> tag contains a summary of which configurations this solution definition performs. The summary is displayed in the Summary tab of the GAX user interface.

Add the <profile> element.

Add the <profile> element by using the following code as an example:

Important

If profile name="install", the profile installs a solution. If profile name="update", the profile updates a solution from a previous version.

[+] Click here to reveal code

```
<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
    xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg"
    xmlns:fileio="http://genesys.com/gax/asd/fileio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
    http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd">

    <deploymentkey>HostedProviderEdition8.1.3</deploymentkey>
    <name>DemoTest</name>
    <version>1.1.1</version>
    <description>This is the demo service package.    This service package is
```

```
intended for testing the GAX automated deployment feature. It will perform one pre-
installation task, one post-installation task, and an installation of the SIP Server
application</description>
  <summary>This is the service package definition file to deploy an Demo
solution</summary>

  <profile name="install">

    </profile>
</solutionDefinition>
```

Important

See the [General Elements](#) reference page for more information about the tags that are used in an SPD file.

Add user input dialog boxes.

Add user input dialog boxes by using the following code as an example: **[+]**
Click here to reveal code

```
<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
  xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg"
  xmlns:fileio="http://genesys.com/gax/asd/fileio"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
  http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd">

  <deploymentkey>HostedProviderEdition8.1.3</deploymentkey>
  <name>DemoTest</name>
  <version>1.1.1</version>
  <description>This is the demo service package. This service package is
intended for testing the GAX automated deployment feature. It will perform one pre-
installation task, one post-installation task, and an installation of the SIP Server
application</description>
  <summary>This is the service package definition file to deploy an Demo
solution</summary>

  <profile name="install">

    <dialog step="Tenant and Host Selection">
      <!-- The Management Framework must have a tenant already
created, named something like Tenant_sg02_97_DEMO for example -->
      <input name="TENANT" title="Tenant Name" type="selectOne"
default="" required="true">
        <description>Please select the tenant</description>
        <objectselect>
          <filter value="CfgTenant" name="type"/>
        </objectselect>
      </input>
      <input name="TENANT_HOST_1" title="Tenant Primary Host"
type="selectOne">
        <description>Please select the primary host where the
```

```

tenant applications are to be installed.</description>
      <objectselect>
        <filter value="CfgHost" name="type"/>
      </objectselect>
    </input>
    <input name="SKIP_INSTALL_PKG" title="Skip IP re-installation if
application already installed for the tenant" default="true" type="selectOne"
required="true">
      <description>This parameter specifies whether to force
reinstallation of the IP if the package already exists on the machine for the
tenant.</description>
      <selection>
        <option name="true" value="true"/>
        <option name="false" value="false"/>
      </selection>
    </input>
    <input name="DATA_MODEL" title="Binary Version (32-bit or
64-bit)" default="64" type="selectOne" required="true">
      <description>This parameter defines the 32-bit or the
64-bit version of the binary to be deployed.</description>
      <selection>
        <option name="32" value="32"/>
        <option name="64" value="64"/>
      </selection>
    </input>
  </dialog>
  <dialog step="Application Parameters">
    <input name="agent_no_answer_action" title="Please enter the
value for TServer.agent-no-answer-action" default="demo_test" type="string">
      <description>Please enter the value for TServer.agent-no-
answer-action</description>
    </input>
  </dialog>
</profile>
</solutionDefinition>

```

Dialog tags are displayed on screen in the GUI Deployment Wizard. Dialog boxes allow the user to select the input values that are defined by the input tags. Multiple dialog boxes (Wizard windows) can be created/added. Multiple dialog boxes enable you to group different user-inputs based on their scope.

Important

See the [Dialog Input Elements](#) reference page for more information about the tags that are used in an SPD file.

Add the Execution Block.

The `<execution>` tag is the main tag in the SPD file. It specifies the execution sequence of various actions that are classified in the following groups:

- `<cfg:...>` -- Configuration actions
- `<os:...>` -- OS actions
- `<fileio:...>` -- File I/O actions
- `<http:...>` -- Web services actions

In addition to these actions, the `<execution>` tag supports additional child elements that enable the implementation of various functions:

- `<script>`-Supports Java Script code (See [JavaScript Usage](#) or [JavaScript Representation of Configuration Objects](#) for more information.)
- `<foreach>`-Supports foreach loops
- `<if>`-Supports conditional statements

Add the Execution Block by using the following code as an example:

[+] Click here to reveal code

```
<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
  xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg"
  xmlns:fileio="http://genesys.com/gax/asd/fileio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
  http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd">

  <deploymentkey>HostedProviderEdition8.1.3</deploymentkey>
  <name>DemoTest</name>
  <version>1.1.1</version>
  <description>This is the demo service package. This service package is
intended for testing the GAX automated deployment feature. It will perform one pre-
installation task, one post-installation task, and an installation of the SIP Server
application</description>
  <summary>This is the service package definition file to deploy an Demo
solution</summary>

  <profile name="install">
    <dialog step="Tenant and Host Selection">
      <!-- The Management Framework must have a tenant already
created, named something like Tenant_sg02_97_DEMO for example -->
      <input name="TENANT" title="Tenant Name" type="selectOne"
default="" required="true">
        <description>Please select the tenant</description>
        <objectselect>
          <filter value="CfgTenant" name="type"/>
        </objectselect>
      </input>
      <input name="TENANT_HOST_1" title="Tenant Primary Host"
type="selectOne">
        <description>Please select the primary host where the
tenant applications are to be installed.</description>
        <objectselect>
          <filter value="CfgHost" name="type"/>
        </objectselect>
      </input>
      <input name="SKIP_INSTALL_PKG" title="Skip IP re-installation if
application already installed for the tenant" default="true" type="selectOne"
required="true">
        <description>This parameter specifies whether to force
reinstallation of the IP if the package already exists on the machine for the
tenant.</description>
        <selection>
          <option name="true" value="true"/>
          <option name="false" value="false"/>
        </selection>
      </input>
    </dialog>
  </profile>
</solutionDefinition>
```

```

        </selection>
    </input>
    <input name="DATA_MODEL" title="Binary Version (32-bit or
64-bit)" default="64" type="selectOne" required="true">
        <description>This parameter defines the 32-bit or the
64-bit version of the binary to be deployed.</description>
        <selection>
            <option name="32" value="32"/>
            <option name="64" value="64"/>
        </selection>
    </input>
</dialog>
<dialog step="Application Parameters">
    <input name="agent_no_answer_action" title="Please enter the
value for TServer.agent-no-answer-action" default="demo_test" type="string">
        <description>Please enter the value for TServer.agent-no-
answer-action</description>
    </input>
</dialog>
<execution>
    <!--preInstallation steps-->
    <os:verifyGDA host="${TENANT_HOST_1.name}"/>
    <!--preInstallation steps end-->

    <!--install package steps-->
    <script>TENANT_NAME = TENANT.name;</script>
    <script>SEGMENT_NAME =
TENANT_NAME.split(/_/g)[1];</script>
    <script>TENANT_NAME_NO_PREFIX =
TENANT_NAME.replace(/^Tenant_/,'');</script>
    <script>SEGMENT_NAME_NO_PREFIX = SEGMENT_NAME.replace(/^sg/, '');
</script>
    <script>TENANT_NAME_NO_PREFIX_HOST =
TENANT_NAME_NO_PREFIX.replace(/_gi, '-');</script>
    <script>TENANT_NAME_NO_PREFIX_HOST =
TENANT_NAME_NO_PREFIX_HOST.replace(/^.+?-/,'');</script>
    <script>TENANT_ID = parseInt(TENANT_NAME.split(/_/g)[2]);
</script>

    <script>var TENANT_SIPS_PR = 'TSrvSIP_' + TENANT_NAME_NO_PREFIX
+ '_1_pr' ; </script>
    <script>var TENANT_STATSERV_PR = 'StatServer_' +
TENANT_NAME_NO_PREFIX + '_1_pr' ; </script>

    <script>var SIP_SERVER_APPPORT = 10000 + TENANT_ID ; </script>
    <script>var TENANT_STATSERV_APPPORT = 10800 + TENANT_ID ;

</script>

    <script>var SIP_SERVER_APPPORT = SIP_SERVER_APPPORT.toString()
</script>
    <script>var TENANT_STATSERV_APPPORT =
TENANT_STATSERV_APPPORT.toString() ;</script>
    <script>var INSTALL_BASE_PATH = '/home/genesys/GCTI/' +
TENANT_NAME ; </script>

    <os:installPackage appName="${TENANT_SIPS_PR}"
host="${TENANT_HOST_1.name}" folder="Applications/
Segment_${SEGMENT_NAME_NO_PREFIX}/Tenants_${SEGMENT_NAME}/${TENANT_NAME}" longName="SIP
Server" version="8.1.0" tenantdbid="${TENANT.dbid}"
skipIfInstalled="${SKIP_INSTALL_PKG}" debug="true" appPort="${SIP_SERVER_APPPORT}"
overrideTemplate="TSrvSIP">
        <list name="silentInstall">
            <property section="IPCommon" key="InstallPath"

```

```

value="${INSTALL_BASE_PATH}/TSrvSIP"/>
                                <property section="IPCommon" key="DataModel"
value="${DATA_MODEL}"/>
                                </list>
                                </os:installPackage>
                                <os:installPackage appName="${TENANT_STATSERV_PR}"
host="${TENANT_HOST_1.name}" folder="Applications/
Segment_${SEGMENT_NAME_NO_PREFIX}/Tenants_${SEGMENT_NAME}/${TENANT_NAME}" longName="Stat
Server" version="8.1.0" tenantdbid="${TENANT_NAME}" overrideTemplate="StatServer"
skipIfInstalled="${SKIP_INSTALL_PKG}" debug="true" appPort="${TENANT_STATSERV_APPPORT}">
                                <list name="silentInstall">
                                <property section="IPCommon" key="InstallPath"
value="${INSTALL_BASE_PATH}/StatServer"/>
                                <property section="IPCommon" key="DataModel"
value="${DATA_MODEL}"/>
                                </list>
                                </os:installPackage>
                                <!--install package steps end-->
                                <!--postInstallation steps-->
                                <cfg:readObject name="${TENANT_STATSERV_PR}"
result="TENANT_STATSERV_PR" type="CfgApplication"/>
                                <cfg:updateObject name="${TENANT_SIPS_PR}" type="CfgApplication">
                                <list name="options">
                                <property section="TServer" key="agent-no-answer-
action" value="${agent_no_answer_action}"/>
                                </list>
                                </cfg:updateObject>
                                <cfg:updateObject name="${TENANT_SIPS_PR}" type="CfgApplication">
                                <list name="appservers">
                                <conninfo
appserverdbid="${TENANT_STATSERV_PR.dbid}" connprotocol="addp" transportparams=""
appparams="" mode="CFGTMBoth" timeoutlocal="10" timeoutremote="10"/>
                                </list>
                                </cfg:updateObject>

                                <!--postInstallation steps end-->
                                </execution>
                                </profile>
</solutionDefinition>

```

Or, refer to the following sample for code using web services:

[+] Click here to reveal code

```

<solutionDefinition version="2.0.0" xmlns="http://genesys.com/gax/asd"
    xmlns:os="http://genesys.com/gax/asd/os" xmlns:cfg="http://genesys.com/
gax/asd/cfg" xmlns:http="http://genesys.com/gax/asd/http"
    xmlns:fileio="http://genesys.com/gax/asd/fileio"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://genesys.com/gax/asd asd.xsd
http://genesys.com/gax/asd/os os.xsd
    http://genesys.com/gax/asd/cfg cfg.xsd http://genesys.com/gax/asd/fileio
fileio.xsd http://genesys.com/gax/asd/http http.xsd">

    <deploymentkey>HttpTest</deploymentkey>
    <name>HttpTest</name>
    <version>1.1.1</version>
    <description>This is the demo service package. This service package is
intended for testing the GAX SPD Web access feature.</description>
    <summary>This is the service package definition file to deploy an Demo
solution</summary>

```

```

    <profile name="install">
        <dialog step="Login Parameters">
            <input name="BaseURL"
                title="GAX base URL" hidden="false"
default="http://localhost:8080/gax" type="string">
                <description>Please enter the GAX URL</description>
            </input>
            <input name="Username"
                title="Management Framework User Account" hidden="false"
default="default" type="string">
                <description>Please enter the Genesys Management
Framework username</description>
            </input>
            <input name="Password"
                title="Management Framework User Password"
hidden="false" default="" type="password">
                <description>Please enter the Genesys Management
Framework user password</description>
            </input>
        </dialog>
        <execution>
            <script>
                var login = new Object();
                login.username = Username;
                login.password = Password;
                login.newPassword = "";
                login.newPasswordConfirm = "";
            </script>
            <http:request method="POST" url= "${BaseURL} + '/api/session/
login/'}">
                <header name="Content-Type" value="application/json"/>
                <header name="Accept" value="application/json"/>
                <body>${JSON.stringify(login)}</body>
            </http:request>
            <script>
                var parameter = new Object();
                parameter.name = "CreatedBySpd";
                parameter.keyName = "";
                parameter.helpText = "";
                parameter.type = "STRING";
                parameter.optional = false;
                parameter.global = false;
            </script>
            <http:request method="POST" url= "${BaseURL} + '/api/opm/
parameters/'}" result="RESULT">
                <body>${JSON.stringify(parameter)}</body>
            </http:request>
            <!-- for debugging -->
            <script> log(' RESULT.code :'+ RESULT.code );</script>
            <script> log(' RESULT.location :'+ RESULT.location );</script>
            <!-- for debuggin end -->
            <http:request method="GET" url="${RESULT.location}"
result="RESULT">
            </http:request>
            <!-- for debugging -->
            <script> log(' RESULT.code :'+ RESULT.code );</script>
            <script> log(' RESULT.content :'+ RESULT.content );</script>
            <!-- for debuggin end -->

```

```

                                <script>
                                    var pgt = new Object();
                                    pgt.name = "PGTCreatedBySpd";
                                    pgt.description = "";
                                    pgt.pgtSections = new Array();
                                    pgt.pgtSections[0] = new Object();
                                    pgt.pgtSections[0].key = "main";
                                    pgt.pgtSections[0].displayName = "Main";
                                    pgt.pgtSections[0].parameters = new Array;
                                    pgt.pgtSections[0].parameters[0] =
RESULT.content.id;
                                </script>

                                <http:request method="POST" url= "${BaseURL} + '/api/opm/pgts/'}"
result="RESULT">
                                    <body>${JSON.stringify(pgt)}</body>
                                </http:request>

                                <!-- for debugging -->
                                <script> log(' RESULT.code :'+ RESULT.code );</script>
                                <script> log(' RESULT.location :'+ RESULT.location );</script>
                                <!-- for debuggin end -->

                                <!--postInstallation steps end-->
                                </execution>
                            </profile>
                        </solutionDefinition>

```

Warning

Executing the above code gets your password flushed.

See the [Executable Elements](#) reference page or the [Web Services Elements](#) reference page for more information about the tags that are used in an SPD file.

Save the Solution Definition file that you created as <filename>.XML.

Launch Genesys Administrator Extension.

Navigate to the Solution Definitions panel.

In the Solution Definitions panel, click +.

In the New Solution Definition panel that is displayed to the right, click Browse.

Find the Solution Definition file that you created and select it.

Click Upload. The name of the Solution Definition file, as defined in the Solution Definition, is displayed in the list.

Verify the Solution Definition file is deployable.

To verify that the Solution Definition file can be deployed, perform the following steps:

- a. In the list of Solution Definition files, click the Solution Definition file that you imported in the previous steps.
- b. A related menu is displayed to the right of the Solution Definition list. Verify that the Solution Definition file can be deployed by selecting **Check IP Availability**.
 - The list of required IPs that are found in the IP Repository and those that are missing are displayed in the IP Availability area.
- c. Upload missing IPs to the IP repository before deploying this Solution Definition file.

If there are no missing IPs, click the Deployable check box.

Click Sav to save your changes or Cancel to leave the Solution Definition file unchanged.

Complete SPD samples

READ THIS FIRST: [readme.txt](#)

- [Tenantcreation.xml](#) This SPD creates the Genesys Tenant and the Administrator (Person) that manages this tenant.
- [eservices.xml](#) This SPD contains Install-, Enable-, and Disable-profiles. The Install-profile deploys the eServices Chat Server, Contact Server, Classification Server, E-mail Server, Training Server, Application

Clusters, Stat Server, Genesys Desktop, third-party components, and 8.1.0 and 8.0.2 applications to the tenant. It also makes the necessary configurations on the segment-level applications that are required by eServices. The Enable-/Disable-profiles enable/disable this service by establishing/removing the component connections from the tenant level to the segment level.

- **gvphpe.xml** This SPD deploys the Genesys Voice Platform Service for Genesys Hosted Provider Edition.
- **http.xml** This SPD demonstrates the use of the `<http:request>` tag.
- **inboundvoice.xml** This SPD contains Install-, Enable-, and Disable-profiles. The Install-profile deploys SIP Server, Genesys Desktop, and Stat Server applications to the tenant. It also makes the necessary configurations on the segment-level applications that are required by the inbound voice service. The Enable-/Disable-profiles will enable/disable this service by adding/removing the tenant entry from the network SIP Server's routing strategy list object.
- **outboundvoice.xml** This SPD contains Install-, Enable-, and Disable-profiles. The Install-profile deploys SIP Server, Genesys Desktop, and Stat Server applications to the tenant. It also makes the necessary configurations on the segment-level applications that are required by the outbound voice service. The Enable-/Disable-profiles will enable/disable this service by establishing/removing the component connections from the tenant level to the segment level.
- **tserverAvayahpe.xml** This SPD deploys the Genesys T-Server for Avaya CM for Genesys Hosted Provider Edition.
- **tserverCiscohpe.xml** This SPD deploys the Genesys T-Server for Cisco UCM for Genesys Hosted Provider Edition.
- **webapihpe.xml** This SPD deploys the Genesys Web API Server for Genesys Hosted Provider Edition.
- **wfm.xml** This SPD contains Install-, Enable-, and Disable-profiles. The Install-profile makes the necessary configuration updates on the database and WFM system to deploy the WFM service. The Enable profile re-enables a previously disabled WFM service by enabling the Database Access Point Objects configured on the tenant level. The Disable-profile disables a previously deployed WFM service by disabling the tenant's user access to the WFM applications, as well as removes the users from the WFM system.  Note: If a service type that contains SIP Server is deployed later, it is necessary to run the WFM re-enable service to properly configure the SIP Server connection for WFM.

References

The Automated Solution Deployment capability within GAX is guided by the XML markup in the solution definition XML file. The following wiki pages outline the elements, attributes, and behavior of the XML markup. The **<solutionDefinition>** element is the root element of a typical solution definition XML file.

- [General Elements](#)
- [Dialog Input Elements](#)
- [Flow Control Elements](#)
- [Executable Elements](#)
- [Web Services Elements](#)
- [Configuration Object Manipulation Elements](#)
- [File I/O Elements](#)
- [Operating System Manipulation Elements](#)

- [Workforce Management Elements](#)
- [Enumerated Types](#)
- [JavaScript Usage](#)
- [JavaScript Representation of Configuration Objects](#)