**Administration Guide | PUBLIC**
Document Version: 8.00 – 2022-11-23

# SAP UI Landscape Configuration Guide

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN SAP

# Content


**1 Getting Started. . . . . 3**

1.1 Configuration Files. . . . . 3

Centrally Managed Configuration. . . . . 3

Local Configuration Files. . . . . 4

Configuration Landscape Overview. . . . . 4

1.2 Configuration Data Migration. . . . . 5

**2 Creating, Displaying and Editing SAP UI Landscape Data. . . . . 6**

2.1 SAP UI Landscape Maintenance Tool. . . . . 6

**3 SAP UI Landscape Format XML Description. . . . . 7**

3.1 SAP UI Landscape Format Example. . . . . 7

3.2 SAP UI Landscape Format Specification. . . . . 10

Tag Structure Overview. . . . . 10

Special Attribute Type. . . . . 10

Header Attributes. . . . . 11

Services Repository. . . . . 11

Workspace Repository. . . . . 20

Message Servers Repository. . . . . 23

Routers Repository. . . . . 24

LDAP Repository. . . . . 24

Webdispatchers Repository. . . . . 25

Includes Repository. . . . . 26

Configuration Repository. . . . . 27

**4 Data Creation with the Java Command Line Tool. . . . . 28**

**5 SPECIAL "LSAdmin" TRACE. . . . . 31**

# 1 Getting Started

As of SAP GUI for Java and Windows 7.40, the landscape configuration has been simplified with the help of a new file format. This new format is called “SAP UI Landscape” and has the following advantages:

- **Easier Configuration**: The persisted connection and connection configuration data from SAP GUI for Windows, SAP GUI for Java and SAP Business Client (SAP BC) are unified in a modern and easy way:
  - Configuration files can either be stored locally or on a central HTTP server or share.
  - The global configuration files are the same for SAP GUI for Windows, SAP GUI for Java and SAP Business Client.
- **Automatic Data Migration**: If SAP GUI has been used already, the connection data will be migrated automatically into the SAP UI Landscape format.

> **i** Note
>
> HTTPS access in SAP GUI for Windows is supported as of SAP GUI for Windows 7.50 patchlevel 6.

## 1.1 Configuration Files

There can be more than one source used for data querying. Number and management of sources depend on the specific implementation, but normally there are at least two sources: companywide, defined by a domain administrator, for example, and a local one, containing services and workspaces defined by the user.

### 1.1.1 Centrally Managed Configuration

As of SAP GUI for Java and Windows 7.40, information formerly stored in separate files for message servers, routers, system descriptions, etc., can be stored in one single Landscape file. This is also valid for custom connection entries that were stored in the connection file before. Used as a central Landscape file, SAP systems relevant to all users can be included in this file, so that they are available at all front-end computers. The SAP UI Landscape format is used as default as of SAP GUI for Java 7.40 and as of SAP GUI for Windows 7.50.

You can create and edit SAP UI Landscape XML data with the SAP UI Landscape Maintenance Tool which can be either accessed via transaction SLMT or by starting report RSLSMT. For details, see section SAP UI Landscape Maintenance Tool [page 6].

Another option is the automatic conversion with the administrator command line tool in Java. For details, see Data Creation with the Java Command Line Tool [page 28].

## 1.1.2 Local Configuration Files

The SAP GUI for Windows and Java stores all of its local configuration information and user preferences in files.

The following information is stored in the `SAPUILandscape.xml` file:

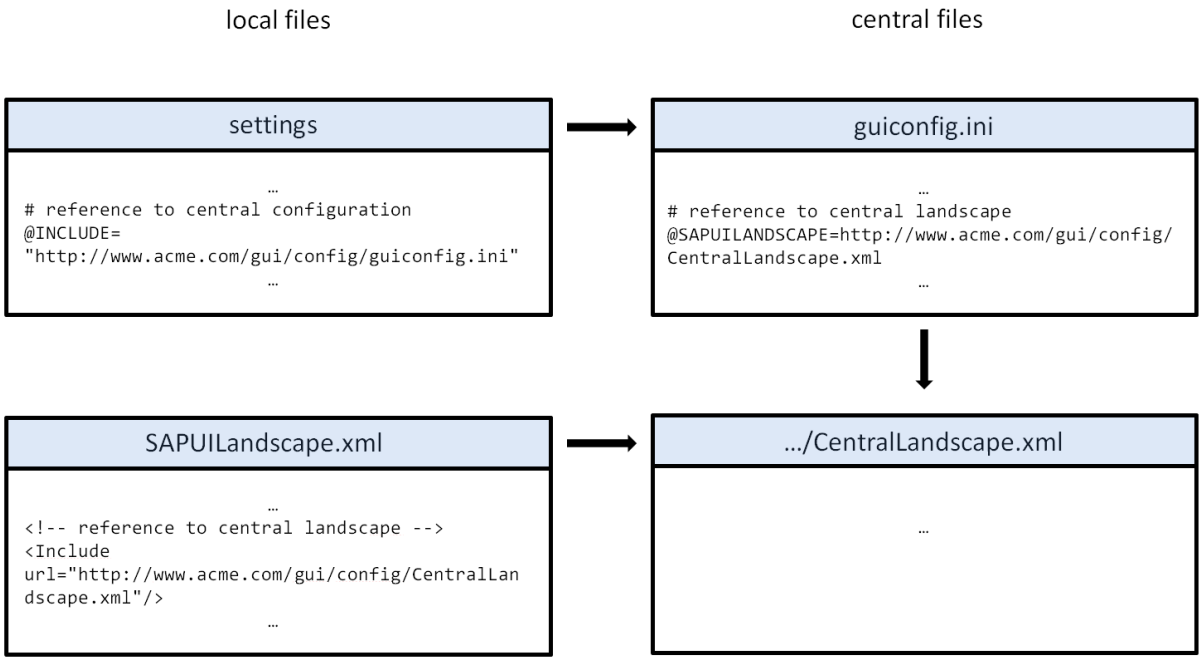
- Connections to SAP systems defined by the user
- Hierarchical structure on the SAP Logon items when using hierarchical view
- User specific notes attached to a connection string
- System descriptions
- Message server entries
- Router entries
- URLs for getting a system status of an SAP system

These local configuration files are created empty when SAP GUI is started for the first time, or they import the values stored in the old configuration files (if available).

> **i Note**
>
> Please note, that this import only happens once.

## 1.1.3 Configuration Landscape Overview

The following picture shows a landscape overview based on a SAP GUI for Java example.



## 1.2 Configuration Data Migration

If SAP GUI for Java or Windows has been used already, the data will be migrated automatically into the new landscape files with the first start of SAP GUI.

> **i** Note
>
> For SAP GUI for Java, only connections stored in the local files `connections` and `connectionTree.XML` are migrated. For SAP GUI for Windows, only `saplogon.ini` and `SAPLogonTree.xml` are migrated.

# 2 Creating, Displaying and Editing SAP UI Landscape Data

The SAP UI Landscape format is an XML format. You find an example under SAP UI Landscape Format Example [page 7].

There are several ways to create and work with SAP UI Landscape data:

- Using the SAP UI Landscape Maintenance Tool is the easiest way to create, display and edit landscape data. See SAP UI Landscape Maintenance Tool [page 6].
- Another option to create SAP UI Landscape data is the automatic conversion with the Java administrator command line tool. See Data Creation with the Java Command Line Tool [page 28].
- In SAP GUI for Windows, you can also use SAP Logon to create a SAPUILandscape.xml that you can distribute to the end users, see SAP Note 2075073 for more information.

## 2.1 SAP UI Landscape Maintenance Tool

You can create, display and edit SAP UI Landscape XML data centrally with the SAP UI Landscape Maintenance Tool which can be either accessed via transaction **SLMT** or by starting report **RSLSMT**. With this tool, XML data can be persisted in the database. You find more information on the functionality in the system documentation of the corresponding transaction or report (i-button).

> **i Note**
>
> - You have to implement note 2311166 to be able to use the report/transaction.
> - You need the corresponding authorization:
>   - role **SAP_SLMT**
>   - authorization object **S_LSMT** with the following values:
>     - **02** for change authorization
>     - **03** for display authorization

# 3 SAP UI Landscape Format XML Description

This chapter describes the SAP UI Landscape XML format. In the first section, you find an example with a short description as general overview. The other sections in this chapter contain a detailed description of the xml format.

> **i** Note
>
> If you make manual changes to configuration files, it is recommended that you validate these files in accordance with the procedure described in note 2112449. However, the validation of the XML file cannot assess the content; it only checks whether the structure of the document is correct.

## 3.1 SAP UI Landscape Format Example

This section contains an example for a SAP UI Landscape XML file. The following picture below shows the basic elements of the xml file. Under SAP UI Landscape Format Specification, you find the detailed specification of all attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Landscape xsi:noNamespaceSchemaLocation="SAPUILandscape.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1" updated="2017-03-07T16:04:14Z" origin="http://acme.com/config/SAPUILandscape.xml" generator="SAP GUI for Java 7.50">
  - <Configuration>
      <Parameter value="http://acme.com/systemstatus" key="SystemStatusURL"/>
    </Configuration>
  - <Includes>
      <Include url="http://acme.com/config/SAPUILandscapeIT.xml"/>
    </Includes>
  - <Services>
      <Service type="SAPGUI" mode="1" server="appserv.acme.com:64001" msid="" name="ABC [appserv.acme.com]" uuid="eda7e634-8482-47e8-bd2e-094e7fbf90c0"/>
    - <Service type="SAPGUI" server="PUBLIC" msid="8877f52f-8628-41a7-b1aa-770f9023ad14" name="ABC [PUBLIC]" uuid="e1ab714e-50da-4cba-9454-5816d332d389">
        <Memo xml:space="preserve">This is a Memo for a service with several lines _____ </Memo>
      </Service>
      <Service type="Reference" name="ABC [PUBLIC] Reference" uuid="2cf6949e-bb58-49dd-a3c5-4939744dcc08" link="e1ab714e-50da-4cba-9454-5816d332d389"/>
      <Service type="SAPGUI" server="PUBLIC" msid="8877f52f-8628-41a7-b1aa-770f9023ad14" name="ABC [PUBLIC] SSO" uuid="239bddc2-19b8-4b71-a0f5-41327be3b8bd" sncop="9"/>
      <Service url="http://ABC.acmde.com/webdynpro/dispatcher/sap.com/tc~wd~samples~testsuite~uuie/TestSuiteUUIE" type="NWBC" name="NWBC connection" uuid="60631fcf-c526-473b-aa4c-7653253aa79f" systemid="ABC" description="ACME Webdynpro connection to TESTSUITE UUIE"/>
      <Service type="Reference" name="NWBC connection Reference" uuid="b2ba5ffa-e057-4693-9230-e6fbceec5fc5" link="60631fcf-c526-473b-aa4c-7653253aa79f"/>
      <Service url="http://acmde.com/webdynpro/dispatcher/sap.com/tc~wd~samples~testsuite~uuie/TestSuiteUUIE" type="WDA" name="WDA connection" uuid="77a7233d-4867-4d5e-8fc1-2bba2523fb17" description="ACME Webdynpro connection to TESTSUITE UUIE"/>
    </Services>
- <Workspaces default="73d7a3c8-89fc-408e-ab2c-8a0beee169a2">
  - <Workspace name="Default Workspace TEST 3" uuid="73d7a3c8-89fc-408e-ab2c-8a0beee169a2" description="The default workspace created by TEST 3" expanded="1">
    - <Node name="NetWeaver Business Client" uuid="9ee6daeb-64b5-4ede-ad02-8a878f56fd24">
        <Item uuid="af8f5c85-276b-4565-ad47-837ec1af00a3" serviceid="60631fcf-c526-473b-aa4c-7653253aa79f"/>
        <Item uuid="fc9b539b-541d-499b-8afa-8266b888d808" serviceid="b2ba5ffa-e057-4693-9230-e6fbceec5fc5"/>
      </Node>
    - <Node name="Web AS ABAP" uuid="71498e51-0951-4bc3-9f8d-a059fbe1e2a2">
        <Item uuid="32fb33d3-3659-4c0a-910d-cd5982f7a794" serviceid="239bddc2-19b8-4b71-a0f5-41327be3b8bd"/>
        <Item uuid="3d6fe003-61f8-4510-82c6-e0d20bbabc22" serviceid="e1ab714e-50da-4cba-9454-5816d332d389"/>
        <Item uuid="8c51b4f0-cefc-497d-a0a9-cb54b6bdb883" serviceid="eda7e634-8482-47e8-bd2e-094e7fbf90c0"/>
        <Item uuid="e84a1345-10e4-410e-a3fb-a4709ae9939f" serviceid="2cf6949e-bb58-49dd-a3c5-4939744dcc08"/>
      </Node>
    - <Node name="Webdynpro" uuid="cd42e0bb-1026-4b2f-8dd0-e89faff5a9d0">
        <Item uuid="ab350d1e-ccb6-4678-8e37-a32d4ed8ba65" serviceid="77a7233d-4867-4d5e-8fc1-2bba2523fb17"/>
      </Node>
    </Workspace>
  </Workspaces>
- <Messageservers>
    <Messageserver name="ABC" uuid="8877f52f-8628-41a7-b1aa-770f9023ad14" description="Messageserver to acme DOT com" port="3201" host="abc.acme.com"/>
  </Messageservers>
- <Routers>
    <Router name="Router" uuid="ac84a5ea-1737-40e6-b39e-e7a8d1ce08af" description="Router for ACME Corp." router="/H/router.acme.com/S/1234"/>
  </Routers>
</Landscape>
```

The file consists of the following basic elements:

1. Specification of the encoding at the top.
2. Definition of the message servers available in the current system landscape with `uuid`, `name`, `host` and `port`.

> **i Note**
>
> The easiest way to create a uuid is to use the SAP UI Landscape Maintenance Tool. There, uuids are created automatically.
>
> And of course, you also find the resources to generate uuids in the internet. On UNIX-based systems, for example, you can generate UUID strings with the command line tool *uuidgen*.

This message server list should contain at least the message server definitions used in the *<services>* section described below. Instead of defining the message servers here, you can also refer to them and other data via includes (see bullet 6 below).

3. Definition of the routers with **`uuid`**, **`name`** and complete **`router string`**.
4. Definition of services
   This is a flat, non-hierarchical list of all services referred to from within Workspaces. The list can contain also services not referred to that are used on client side for creating user workspaces or for building other functionality such as a favorite list or a search provider list.
   A service contains all data necessary to connect to a SAP system via SAP GUI. Generally, a service entry consists of **`uuid`**, **`name`** and **`type`** (= service type, for example, SAP GUI connection, SAP GUI shortcut, SAP Business Client connection to WebAS ABAP or Portal, or search provider used by SAP Business Client). Depending on the service type, you specify additional attributes (see specification).
5. Definition of workspaces and its nodes and items
   A workspace is a group of connections for certain users, for example, for information developers. It consists of a group of services (items) and folders (nodes) and can be used to show some predefined service set in UI or to do multi-logon for all services in the workspace. The workspace is so-to-speak a special kind of top node.
   The nodes (folders) are item grouping elements of the workspace. They are used to organize the items (= services) within the workspace. From nodes and items, tree structures can be built. Items are placeholders for services and the data is taken from the service description. You can define items directly in the workspace or in a node.
6. Insertion of includes
   An include is a file or URL containing information that you want to merge with the content in your XML. This means, for example, that you do not need to define message servers directly in this XML file, but that you can point to this information from within the includes section. An include node describes a single source to include.
7. System check via configuration
   The **`Configuration`** repository contains a list of parameters related to the system landscape. It has no own attributes. With the data under **`Configuration`**, the server status can be checked. Currently, two parameter keys are supported: **`SystemStatusURL`** and **`SingleSystemStatusURL`**. In case no connection can be established to the backend/server, the system status can still be checked via these two web sites. The first parameter holds a generic status page URL, the second a query URL that contains a **`%s`** parameter which is replaced with a specific systemid.

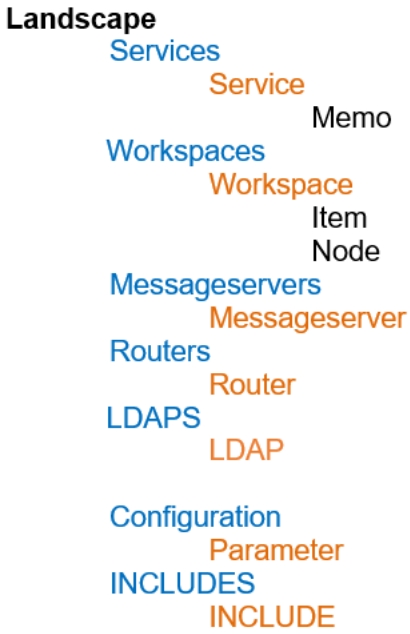
> **i Note**
>
> You can insert regular XML comments.
>
> If a section remains empty, for example the **`<Routers>`** section, it has not effect.

## 3.2 SAP UI Landscape Format Specification

This section contains the detailed description of the XML format.

### 3.2.1 Tag Structure Overview

This overview shows the structure of the tags inside the SAP UI Landscape XML format. The repositories are shown in blue, their nodes in orange.




### 3.2.2 Special Attribute Type

The following list shows the special attribute types that are used in this specification.

| Attribute | Description |
| --- | --- |
| *uuid* | Immutable, universally unique identifier (UUID). A UUID represents a 128-bit value. For more information including algorithms used to create UUIDs, see RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace , section 4.2 "Algorithms for Creating a Time-Based UUID".<br>Example: "**d5bf6876-0ee9-4ae2-8c68-9aeb07081a5e**" |

| Attribute | Description |
| --- | --- |
| *boolean* | A boolean value with "0"=false or "1"=true, if not otherwise specified. |
| *long* | Used for timestamp: number of seconds since January 1, 1970. |
| *service type* | Known Service Types such a NWBC, SAPGUI, etc. You find a list of the service types and their attributes in section Service-Type Specific Attributes [page 12]. |

## 3.2.3 Header Attributes

The root node of the xml is a Landscape tag:

| Attribute | Description |
| --- | --- |
| *version* | Version of the landscape format; currently: 1 |
| *origin* | Original location (source) of the current landscape as URL. |
| *updated [optional]* | Timestamp of the last landscape change with the format “yyyy-MM-dd'T'HH:mm:ssZ”, which represents the time in UTC. |

A Landscape has two main sections: Services and Workspaces. These are described in the next to sub sections.

> **i Note**
>
> The data type of the attributes is string, if not specified otherwise.

## 3.2.4 Services Repository

The Services Repository is a flat, non-hierarchical list of all services referred to from within *Workspaces*. The list can contain also services not referred to that are used on client side for creating user workspaces or for building other functionality such as favorite list or search provider list.

### 3.2.4.1 General Attributes

Basically, a service entry consists of the following general attributes:

| Attribute | Description |
|---|---|
| *uuid*<br>*[mandatory, uuid]* | Unique ID of the service, to be referred to later for unique identification during merge with user (client-defined) services, see also Special Attribute Type [page 10]. |
| *name*<br>*[mandatory]* | Name of the service to be displayed in service list |
| *type*<br>*[mandatory]* | Service type attribute (see list of allowed type IDs below) |
| *description*<br>*[optional]* | If not provided, name is used as default value. |
| *memo*<br>*[optional]* | If the description attribute is not sufficient, every service type can optionally have a *Memo* node inside, containing free-form, multiline text. Trimming of the white space characters by different browsers should be considered when rendering the spec XML with xml:space="preserve".<br>Sample Code<br>`<Service type="SAPGUI" uuid="d5bf6876-0ee9-4ae2-8c68-9aeb07081a5e" name="B30"...>`<br>`<Memo xml:space="preserve">`<br>`Free text…..`<br>`</Memo>` |

### 3.2.4.2 Service-Type Specific Attributes

This section contains the detailed description of the different service types and their attributes.

#### 3.2.4.2.1 SAP GUI

In the following list, you find the additional attributes of the SAP GUI service type and their possible values. And below the table, there are also some SAP GUI Service connection examples listed.

| Area | Attribute | Description |
| --- | --- | --- |
| **Host** | *mode* | Mode for host specification. Possible values:<br>• **0** – group message server. For load-balancing with a message server and logon group (see the old **"/R/messageserver/G/group"** or **"/M/mshost/S/msport/G/group"**).<br>• **1** – application server. For direct application server mode (see the old **"/H/applicationserverhost/S/port"**).<br>• **2** – GUIParam (GUI Parameters). For Windows GUI legacy mode with **"guiparam"**. |
| | *msid*<br>*[optional, uuid]* | Message server ID from the message server repository. |
| | *server*<br>*[optional]* | Fully qualified application server address (host + port) or logon group depending on the *mode* value.<br>Examples: **"abcd.acme.com:3206"**, **"PUBLIC"**. |
| | *routerid*<br>*[optional, uuid]* | Router ID from the router ID repository |
| **SNC** | *sncop*<br>*[optional, int]* | SNC operation<br>Possible values:<br>• **0** (disabled)<br>• **1** (AUTHENTIFICATION)<br>• **2** (NTEGRITY)<br>• **3** (ENCRYPTION)<br>• **9** MAXIMUM<br>• **-1** (deprecated, for backward compatibility only) |
| | *sncname*<br>*[optional]* | SNC Name<br>Example: **"p/secude:CN=ACM, O=ACME, C=DE"**. |
| | *sncnosso*<br>*[optional, boolean]* | No SSO, secure connection only |
| **Encoding** | *encoding*<br>*[optional]* | Protocol encoding. Values are: **"diag"**, **"cdiag"**, **"R3xml"** |
| | *protocol*<br>*[optional]* | Transport protocol. Values are: **"ni"**, **"R3http"** |

| Area | Attribute | Description |
|---|---|---|
| **Content Encoding** | *sapcpg*<br>*[optional, int]* | SAP Codepage<br>Examples: `0`, `1`, `2`, `1100` etc. |
| | *uncoff*<br>*[optional, boolean]* | Flag to disable Unicode. Backward compatibility. |
| | *cencoding*<br>*[optional]* | Encoding. Backward compatibility.<br>Examples: `"UTF-8"`, `"ISO-8859-1"` |
| | *clocale*<br>*[optional]* | Locale. Backward compatibility.<br>Examples: `"de"`, `"en"`, `"en_us"`, etc. |
| **Upload/Download** | *dcpg*<br>*[optional]* | SAP Code page for upload/download<br>Examples: `0`, `1`, `2`, `1100` etc. |
| **Logon** | *client*<br>*[optional]* | Default client<br>Example: `"000"`, `"100"` |
| | *user*<br>*[optional]* | Default user name |
| | *language*<br>*[optional]* | Logon language |
| | *cmd_type*<br>*[optional, enum]* | Command type<br>Values: `"Transaction"`, `"Report"`, `"SystemCommand"`. |
| | *cmd*<br>*[optional]* | Command; content depends on `cmd_type`. |
| | flavorid<br>*[0..1, string ]* | SAP Screen Personas flavor id that is used for a transaction.<br>Example: 005056AB61161ED5A0C7622B6199D9E5 |
| **Shortcut** | *shortcut*<br>*[optional, boolean]* | Shortcut connection |
| | *reuse*<br>*[optional, boolean]* | Reuse connection |

| Area | Attribute | Description |
|---|---|---|
| | *connid*<br>*[optional]* | ConnectionID |
| | *winmax*<br>*[optional, boolean]* | Window maximized |
| | *systemid*<br>*[optional]* | System name = SAP Database Name = SAP System ID <- for shortcut connection reuse.<br>Example: `"B30"`. |
| | *guiparam*<br>*[optional]* | Legacy GUI command line parameters; shortcut should be set to `1`, mode is `2` for GUIParam. |
| **Misc.** | *wan*<br>*[optional, boolean]* | WAN flag |
| | *wp*<br>*[optional, boolean]* | Workspace flag |
| | *rfcid*<br>*[optional]* | RFC ID reserved for dialog RFC usage |
| | *sso2*<br>*[optional]* | SSO token reserved for dialog RFC usage |

## SAP GUI Service Connection Examples

- *SAPGUI connection with load balancing*: a message server (attribute *msid*) and a logon group (attribute *server*) are needed

  Sample Code

  ```
  Service uuid="e1ab714e-50da-4cba-9454-5816d332d389"
  name="ABC [PUBLIC]"
  msid="8877f52f-8628-41a7-b1aa-770f9023ad14"
  server="PUBLIC"
  type="SAPGUI"/>
  ```

- *SAPGUI connection to an application server*: attribute mode must be set to `1` and an application server (attribute server) must be specified

  Sample Code

  ```
  <Service uuid="eda7e634-8482-47e8-bd2e-094e7fbf90c0"
  ```

```
name="ABC [appserv.acme.com]"
mode="1"
server="appserv.acme.com:64001"
type="SAPGUI"/>
```

- *SSO connection*: set attribute sncop (9 == use maximum available security)

Sample Code

```
<Service uuid="239bddc2-19b8-4b71-a0f5-41327be3b8bd"
name="ABC [PUBLIC] SSO"
msid="8877f52f-8628-41a7-b1aa-770f9023ad14"
server="PUBLIC" sncop="9" type="SAPGUI"/>
```

## 3.2.4.2.2 NWBC

In the following list, you find the additional attributes of the SAP Business Client (SAP BC) service type and their possible values.

| Attribute | Description |
|---|---|
| *url*<br>*[mandatory]* | Service URL |
| *client*<br>*[optional]* | Default client |
| *user*<br>*[optional]* | Default user name |
| *language*<br>*[optional]* | Logon language |
| *msid*<br>*[optional, UUID]* | Message server ID from the message servers repository |
| server<br>*[optional]* | Fully qualified application server address (host + port) or name of web dispatcher from the web dispatcher's repository<br>Example: **"abcd.acme.com:3206"** |
| *slc*<br>*[optional, boolean]* | Usage of SLC (secure login client – NW SSO) for login |

| Attribute | Description |
| --- | --- |
| *Ssoparameter*<br>*[optional]* | SSO parameter<br>Example: `"spnego=disabled"`; `"saml2=disabled"` |
| *systemid*<br>*[optional]* | System name = SAP Database Name = SAP System ID<br>Example: `"B30"` |
| *sapguiid*<br>*[optional, UUID]* | Service ID from service repository for service of type SAP-GUI. Used only to get a name and description of an existing SAPGUI connection. |
| *uses_webdispatcher*<br>*[optional]* | If `uses_webdispatcher="1"`, all web requests of the SAP Business Client are sent to the system via the web dispatcher address in the service URL. |

## 3.2.4.2.3 FIORI

In the following list, you find the additional attributes of the SAP Fiori Launchpad (FIORI) service type and their possible values.

| Attribute | Description |
| --- | --- |
| *url*<br>*[mandatory]* | Service URL |
| *client*<br>*[optional]* | Default client |
| *user*<br>*[optional]* | Default user name |
| *language*<br>*[optional]* | Logon language |
| *msid*<br>*[optional, UUID]* | Message server ID from the message servers repository |
| server<br>*[optional]* | Fully qualified application server address (host + port) or name of web dispatcher from the web dispatcher's repository.<br>Example: `"abcd.acme.com:3206"` |

| Attribute | Description |
| --- | --- |
| *slc*<br>*[optional, boolean]* | Usage of SLC (secure login client – NW SSO) for login |
| *systemid*<br>*[optional]* | System name = SAP Database Name = SAP System ID<br>Example: `"B30"` |
| *sapguitype*<br>*[0..1,string]* | Used to obtain a browser stand-alone behavior for FLP in SAP Business Technology Platform (SAP BTP) systems.<br>Values: “WinGUI” == SAP GUI for Windows, “WebGUI” == SAP GUI for HTML else SAP GUI for Windows with fallback. |
| *uses_webdispatcher*<br>*[optional]* | If `uses_webdispatcher="1"`, all web requests of the SAP Business Client are sent to the system via the web dispatcher address in the service URL. |

## 3.2.4.2.4 Portal

In the following list, you find the additional attributes of the Portal service type and their possible values. The Portal service type is used by SAP BC.

| Attribute | Description |
| --- | --- |
| *url*<br>*[mandatory]* | Service URL |
| *user*<br>*[optional]* | Default user name |
| *language*<br>*[optional]* | Logon language |
| *slc*<br>*[optional, boolean]* | Usage of SLC (secure login client – NW SSO) for login |
| filterid<br>*[0..1*] | Portal filter ID<br>A grouping of specific settings that control what kind of data is displayed in the system you are logging on to. |

## 3.2.4.2.5 Sidepanel

In the following list, you find the attributes of the SIDEPANEL service type and their possible values. The SIDEPANEL service type is used by SAP BC.

| Attribute | Description |
|---|---|
| *url*<br>*[mandatory]* | URL string |

## 3.2.4.2.6 Search

In the following list, you find the attributes of the Search service type and its possible values. The Search service type is used by SAP BC. Search refers to the search service link (such as Google, SAP TREX or Help Search). If parameters is not filled, the parameters string is not used when querying a service.

| Attribute | Description |
|---|---|
| *url*<br>*[mandatory]* | Service URL (*http://www.google.com*) |
| *parameters*<br>*[optional]* | Parameters string with place holders for search terms (for example **"?q={0}"**, where {0} is placeholder for user search input) |
| *mnemonic*<br>*[optional]* | Shorthand symbol that can be selected as search location in Quick Launch. |
| *canvasapptype*<br>*[mandatory]* | canvas application type<br>valid values: **BrowserURL**, **Empty**, **SapguiTransaction**, **BrowserTransaction**, **BrowserBSP**, **BrowserWU**, **BrowserServiceMap**, **BrowserWebDynpro**, **BrowserES** |

## 3.2.4.2.7 Reference

Reference is a special service type which simplifies the reuse of the existing services by overwriting some attributes.

| Attribute | Description |
| --- | --- |
| *link*<br>*[mandatory, uuid]* | ID of other service from within the list, the attributes of which should be overwritten.<br>This service is used by all clients. |

#### 3.2.4.2.8 AO

SAP BusinessObjects Analysis connection to an SAP HANA system

| Attribute | Description |
| --- | --- |
| *scheme*<br>[optional, string] | Scheme for URL; currently „http“ or „https" (default: „http" if attribute is not set) |
| *host*<br>*[mandatory, string]* | Host of the server<br>Example: `"binmain.acme.com"` |
| *port*<br>*[mandatory, int]* | Port to service |
| *msid*<br>*[optional, UUID]* | Message server ID from message servers repository. |
| *servertype*<br>*[optional, int]* | Target server platform<br>`0`=SAP NW http connection, `1`=SAP HANA, `2`=BIP Business Intelligence platform (default= 0 if property is not set) |
| *auth*<br>*[optional, string]* | Authentication method for target platform, see servertype (for details, see SAP BusinessObjects Analysis Office documentation: http://help.sap.com/boaa). |

## 3.2.5 Workspace Repository

A workspace is a group of connections for certain users, for example, for information developers. It consists of a group of services (items) and folders (nodes) and can be used to show some predefined service set in UI or to do multi-logon for all services in the workspace. The workspace is so-to-speak a special kind of top node.

The nodes (folders) are item grouping elements of the workspace. They are used to organize the items (=services) within the workspace. From nodes and items, tree structures can be built. Items are placeholders for services and the data is taken from the service description. You can define items directly in the workspace or in a node.

> **Note**
>
> Server configuration files must not contain a workspace called “Local” due to conflicts with local configuration files.
>
> Therefore, when creating server configuration files via SAP Logon / SAP Business Client / SAP GUI for Java, you need to rename the workspace “Local” before distributing the files

Workspace(s), nodes and items are defined as follows:

| Area | Attribute | Description |
|---|---|---|
| **Workspaces**<br><br>**(Contains a list of workspaces)** | *default optional* | The «Workspaces» section contains a list of workspaces. You can set a default workspace:<br><br>default [optional] – attribute specifying the default workspace ID from within the workspaces list. If not specified, the first workspace is used as default.<br><br>If **Workspaces** section is empty, a default workspace will be assumed.<br><br>**Note**<br><br>1) This feature is not available in SAP GUI for Windows.<br><br>2) This feature can only set a one-time default value. After the user has selected a different workspace, the workspace the user selected will be used as default. |
| **Workspace** | *uuid*<br><br>*[mandatory, UUID]* | mandatory ID of the workspace; see also Special Attribute Type [page 10] |
| | *name*<br><br>*[mandatory]* | Name of the workspace |
| | *description*<br><br>*[optional]* | Description of the workspace |

| Area | Attribute | Description |
| --- | --- | --- |
| | *timestamp*<br>*[optional, long]* | Attribute used on client side to note the time of the last change inside the workspace. The time stamp attribute on workspace level should be only used for "foreign" (global) workspaces. Based on the time stamp, the client application can decide when to purge dangling user modifications for the global workspace or how to resolve merging conflicts. |
| | *hidden*<br>*[optional]* | Marks workspace as hidden and not visible in UI. |
| **Node (Folder)** | *uuid*<br>*[mandatory, uuid]* | mandatory ID of the node; see also Special Attribute Type [page 10] |
| | *name*<br>*[mandatory]* | Name of the folder |
| | *description*<br>*[optional]* | Description of the folder |
| | *expanded*<br>*[optional]* | State of folder in UI: collapsed or expanded. Default value is `'false'` (0=expanded). |
| | *timestamp*<br>*[optional, long]* | Attribute used on client side to note time stamp of a change applied to node by the user. Based on the time stamp, the client application can decide when to purge dangling user modifications, when the node is deleted on server side. |
| | *hidden*<br>*[optional]* | Mark node as hidden and not visible in UI. Allows changing workspace configuration locally. |
| **Item** | *uuid*<br>*[mandatory, UUID]* | mandatory unique ID of the item; see also Special Attribute Type [page 10] |
| | *serviceid*<br>*[mandatory]* | ID of the service this item refers to (should be one of known services from Services repository) |

| Area | Attribute | Description |
| --- | --- | --- |
| | *timestamp*<br>*[optional, long]* | Attribute used on client side to note time stamp of a change applied to item by the user. Based on the time stamp, the client application can decide when to purge dangling user modifications, when the item is deleted on server side. |
| | *hidden*<br>*[optional]* | Marks item as hidden and not visible in the UI. Allows changing workspace configuration locally. |

Normally, it is not possible to delete a service link from a workspace loaded from an external (global) source. To solve this, the *hidden* attribute has been introduced. It marks the current service link as invisible in the UI.

> **i** Note
>
> The functionality to hide and show objects via the User Interface is not available in SAP GUI for Windows.

## 3.2.6 Message Servers Repository

The *Messageservers* repository (collection node on the same level as Services) contains a list of message servers available in the current system landscape. The list should contain at least message server definitions referred to from entries in the current Services repository (see above). Several service entries can point to the same message server.

A message server node describes one message server and can have the following attributes:

| Attribute | Description |
| --- | --- |
| *uuid*<br>*[mandatory, uuid]* | Mandatory ID of the message server, see also Special Attribute Type [page 10]. |
| *name*<br>*[mandatory]* | Name of the message server, usually the system ID of the server. |
| *host*<br>*[mandatory]* | Host of the server<br>For example: `"binmain.acme.com"`. |
| *port*<br>*[mandatory, int]* | Port number to message server service<br>Example: `"3276"`. |

| Attribute | Description |
| --- | --- |
| *description*<br>*[optional]* | Description of the message server |
| *routerid*<br>*[optional, uuid]* | UUID of the router from Routers repository |
| *routerid*<br>*[optional, uuid]* | UUID of the MS LDAP server from the LDAP Repository [page 24]. |

## 3.2.7 Routers Repository

The *Routers* repository (collection node on the same level as *Services*) contains a list of routers available in the current system landscape. The list should contain at least router definitions referred to from entries in the current Services Repository [page 11]. Several service entries can point to the same router.

The *Routers* section contains a list of routers. A router node describes a single router and can have the following attributes:

| Attribute | Description |
| --- | --- |
| *uuid*<br>*[mandatory, uuid]* | Mandatory id of the router, see also Special Attribute Type [page 10]. |
| *name*<br>*[mandatory]* | Name of the router |
| *router*<br>*[mandatory]* | Complete router string<br>Example: `"/H/router01.acme.com/S/6756/P/abc123"`. |
| *description*<br>*[optional]* | Description of the router |

## 3.2.8 LDAP Repository

The repository *LDAPS* (the collection node is one the same level as *Services*) contains the list of LDAP servers available in the current system landscape.

The LDAP node describes single LDAP servers and can have the following attributes:

| Attribute | Description |
|---|---|
| *uuid*<br>*[mandatory, uuid]* | ID of the message server, see also Special Attribute Type [page 10]. |
| *name*<br>*[mandatory]* | Name of the LDAP router |
| *mode* | Mode of the server<br>For example: **`LDAPdirectory`**. |
| *Server*<br>*[mandatory]* | Host address of LDAP server<br>Example: **`wsi6503.nt1.acme.com`**. |
| *node* | |
| *options* | options string for LDAP server<br>Example: **`DirType=NT5ADS`**. |
| *description* | Description of the LDAP server. If no specified name attribute is used. |

## 3.2.9 Webdispatchers Repository

The *Webdispatchers* repository (collection node on the same level as *Services*) contains a list of SAP Web Dispatchers (node *Webdispatcher*) available in the current system landscape.

The *Webdispatcher* node describes one webdispatcher and can have the following attributes:

| Attribute | Description |
|---|---|
| *uuid*<br>*[mandatory, uuid]* | Mandatory id of the web dispatcher, see also Special Attribute Type [page 10]. |
| *name*<br>*[mandatory, string]* | Name of the web dispatcher |
| *description*<br>*[optional, string]* | Description of the web dispatcher. If not specified, the name attribute is used. |
| *url*<br>*[mandatory]* | URL string to access the SAP Web Dispatcher host. |

| Attribute | Description |
| --- | --- |
| *msid*<br>*[mandatory, UUID]* | Message server ID from the Message Servers Repository [page 23] |
| *client*<br>*[optional, string]* | Logon client<br>Example: `"000"`, `"100"` |

### Example

Sample Code

```
<Webdispatchers>
<Webdispatcher uuid="4..." name="Webdispatcher ACME 01" url=https://
webdisp01.acme.com msid ="1..." client=”010”/>
<Webdispatcher uuid="5..." name="Webdispatcher ACME 02" url=https://
webdisp02.acme.com msid ="1..." client=”020”/>
<Webdispatcher uuid="6..." name="Webdispatcher ACME 03" url=https://
webdisp03.acme.com msid ="2...">
</Webdispatchers>
```

## 3.2.10 Includes Repository

An include is a file or URL containing information that you want to merge with the content in your XML. This means, for example, that you do not need to define message servers directly in this XML file, but that you can point to this information from within the *Includes* repository. The repository (collection node on the same level as Services) contains a list of landscape sources to include. An include node describes a single source to include and can have the following attributes:

| Attribute | Description |
| --- | --- |
| *url*<br>*[mandatory]* | URL string to a landscape source |
| *index*<br>*[mandatory, integer]* | Index to import, from lowest to highest |
| *name*<br>*[optional]* | Name of the include |

| Attribute | Description |
| --- | --- |
| *description*<br>*[optional]* | Description of the include |

## 3.2.11 Configuration Repository

The *Configuration* repository contains a list of parameters related to the system landscape. It has no own attributes. With the data under *Configuration* the server status can be checked. Currently, two parameter keys are supported: *SystemStatusURL* and *SingleSystemStatusURL*. As described in SAP Note 1087494 the two support URLs are now configured within the SAP UI Landscape.

In case no connection can be established to the backend/server, the system status can still be checked via these two web sites. *SystemStatusURL* holds a generic status page URL and *SingleSystemStatusURL* a query URL that contains a `%s` parameter which is replaced with a specific systemid.

For one landscape, the keys of the parameters are unique. Local parameters are always preferred. So if a parameter is contained in a “local” landscape and in a “global” landscape, always the local parameter is used as whole. Which means that no individual attributes are merged from “global” to “local” parameter.

The configuration repository should not be used as a mechanism to persist other SAP GUI configurations or settings.

The parameter node has the following attributes:

| Attribute | Description |
| --- | --- |
| *key*<br>*[mandatory]* | Key for the parameter |
| *value*<br>*[mandatory]* | Value of the parameter |

# 4 Data Creation with the Java Command Line Tool

If you use SAP GUI for Windows or Java 7.30 and 7.40 in parallel, you do not need to maintain both configuration files separately. You can maintain the configuration data in either version and convert then the data for use in the other version.

## Prerequisites

- installed SAP GUI for Java 7.40 or later
  `PATH_TO_JARS` is the path to the SAP GUI for Java jar files. By default, the files can be found under:
  UNIX: `[INSTALLATION_DIRECTORY]/SAPGUI7.40/jar/`
  OSX: `[INSTALLATION_DIRECTORY]/SAPGUI 7.40/SAPGUI 7.40.app/Contents/Resources/Java/`
  WINDOWS: `[INSTALLATION_DIRECTORY]\SAPGUI for Java 7.40\jar\`
- JRE 8 (see SAP GUI for Java Documentation for more information)

## Convert File Creation

Place the following command into a script file or a bat file as shown in the examples:

Sample Code

```
--------------------------------------------------------------------------------
Linux bash EXAMPLE
--------------------------------------------------------------------------------
#!/bin/bash
PATH_TO_JARS=/opt/SAPClients/SAPGUI7.40/jar/
if /usr/bin/test "x$PLATIN_JAVA" = "x" ; then
    PLATIN_JAVA=java
fi
$PLATIN_JAVA -Djava.awt.headless=true -jar ${PATH_TO_JARS}/GuiStartS.jar
convert  $@
```

Sample Code

```
--------------------------------------------------------------------------------
-
OS X bash EXAMPLE
--------------------------------------------------------------------------------
-
#!/bin/bash
PATH_TO_JARS="/Applications/SAP Clients/SAPGUI 7.40/SAPGUI 7.40.app/Contents/
Resources/Java/"
JAVAHOME=`/usr/libexec/java_home -v 1.8+ | sed s/\\\n//g`
"${JAVAHOME}/bin/java" -Djava.awt.headless=true -jar "${PATH_TO_JARS}"/
GuiStartS.jar convert  $@
```

Sample Code

```
------------------------------------------------------------------------------
-
Windows bat EXAMPLE
------------------------------------------------------------------------------
-
@echo off
if "%PLATIN_JAVA%"=="" set PLATIN_JAVA=java.exe
set PATH_TO_JARS=%ProgramFiles%\SAP Clients\SAPGUI for Java 7.40\jar
:: Make Java call to the converter.
"%PLATIN_JAVA%" -Djava.awt.headless=true -jar "%PATH_TO_JARS%\GuiStartS.jar"
convert  %*
```

## Conversion Commands

1. **General converter call:**
   *OSX and UNIX:*
   **`java -Djava.awt.headless=true -jar [PATH_TO_JARS]/GuiStartS.jar convert [CONVERTER_PARAMETER]`**
   *WINDOWS:*
   **`java -Djava.awt.headless=true -jar [PATH_TO_JARS]\GuiStartS.jar convert [CONVERTER_PARAMETER]`**
   If you have created a script file with name "converter", you can call the SAP UI Landscape converter as noted in the examples below
2. **Converter help:**
   **`converter -?`**
3. **Basic Commands**
   To create an empty SAPUILandscape.xml file in the output folder:
   **`converter create output/SAPUILandscape.xml`**
   To repair internal problems of XML:
   **`converter repair output/SAPUILandscape.xml`**
4. **Conversion of 7.30 data to 7.40**
   You can maintain your data in the SAP GUI 7.30 version and convert it for use in 7.40. This is done by calling the converter script and the import command as shown in the following examples where “In”/out” refer to the source and destination folder:
   1. Import of SAP GUI for Windows 7.30 configuration data
      **Examples**
      To import SAP GUI for Windows 7.30 messageserver, services and router files from folder input and to write the content into the SAPUILandscape.xml file in the output folder:

      Sample Code

      ```
      converter importW input/sapmsg.ini input/services.unx input/
      saproute.ini input/SAPUILandscape.xml output/SAPUILandscape.xml
      ```

      To convert SAP GUI for Windows 7.30 connection data:

Sample Code

```
converter importConW input/saplogon.ini input/SapLogonTree.xml output/
SAPUILandscape.xml output/SAPUILandscape.xml
```

2. Import of SAP GUI for Java 7.30 configuration data:
   **Examples**
   To convert SAP GUI for Java 7.30 messageserver, message server description and router files:

   Sample Code

   ```
   converter importJ msgPath msgDescriptionPath routerPath SourceLandscape
   TargetLandscape
   ```

   To convert SAP GUI for Java 7.30 connection data:

   Sample Code

   ```
   converter importConJ connectionsFile connectionTreeFile SourceLandscape
   TargetLandscape
   ```

5. **Conversion of 7.40 data to 7.30**
   You can also maintain the data in SAP GUI release 7.40 and export the data to 7.30.
   The commands are the same as in the section above. The only difference is that you have to use **exportW**, **exportConW**, **exportJ** and **exportConJ** as commands.
   **Examples for the Export of SAP GUI for Java configuration data:**
   To convert SAP GUI for Java 7.40 messageserver, message server description and router files:

   Sample Code

   ```
   convert exportJ MDR SourceLandscape msgPath msgDescriptionPath routerPath
   ```

   To convert SAP GUI for Java 7.40 connections:

   Sample Code

   ```
   converter exportConJ SourceLandscape connectionsFile connectionTreeFile
   ```

# 5 SPECIAL "LSAdmin" TRACE

If specified as JavaGUI trace key, a button *Open Landscape* is displayed in the *SAP Logon* window. You can open a `SAP UI Landscape` file, which then can be edited in SAP Logon. See also in the SAP GUI for Java Manual section 5.3.3 'Trace Information'.

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.

THE BEST RUN SAP