



SAP UI Landscape Configuration Guide

SAP GUI for Java, Release 7.50

TABLE OF CONTENTS

1	GETTING STARTED	3
1.1	Configuration Files.....	3
1.1.1	Centrally Managed Configuration.....	3
1.1.2	Local Configuration Files.....	3
1.1.3	Configuration Landscape Overview	4
1.2	Configuration Data Migration.....	4
2	SAP UI LANDSCAPE FORMAT XML DESCRIPTION	4
2.1	SAP UI Landscape Format Example	5
2.2	SAP UI Landscape Format Specification	6
2.2.1	Tag Structure Overview.....	6
2.2.2	Special Attribute Types	7
2.2.3	Header Attributes	7
2.2.4	Services Repository	7
2.2.4.1	<i>General Attributes</i>	<i>7</i>
2.2.4.2	<i>Service-Type Specific Attributes</i>	<i>8</i>
2.2.5	Workspaces Repository	11
2.2.6	Message Servers Repository	12
2.2.7	Routers Repository.....	13
2.2.8	Includes Repository.....	13
2.2.9	Configuration Repository.....	13
3	CONFIGURATION DATA CONVERSION.....	14
3.1	Prerequisites.....	14
3.2	Convert File Creation	14
3.3	Conversion Commands	15
4	SPECIAL "LSAdmin" TRACE	16

1 GETTING STARTED

As of SAP GUI for Java 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 NetWeaver Business Client (NWBC) 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 NWBC.
- **Automatic Data Migration:** If SAP GUI has been used already, the connection data will be migrated automatically into the SAP UI Landscape format.

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 7.40, information formerly stored in separate files for message servers, routers, system descriptions, etc., are now stored in one single Landscape file, as well as the 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.

If you use the centrally managed configuration for SAP GUI for Java 7.30, **WE STRONGLY ADVISE YOU TO PROVIDE A CENTRAL SAP UI LANDSCAPE FOR VERSION 7.40 TOO.** In the section [Configuration Data Conversion](#) it is described how to convert the central 7.30 configuration into the SAP UI Landscape format. Add the directive to @SAPUILANDSCAPE with the url to the central file see ([Configuration Landscape Overview](#)).

If no central SAP UI Landscape is available, see [Configuration Data Migration](#) for further information.

1.1.2 Local Configuration Files

The SAP GUI for Java stores all of its local configuration information and user preferences in files. The files named *SAPGUI_Landscape.xml* are located in a place that is platform-specific:

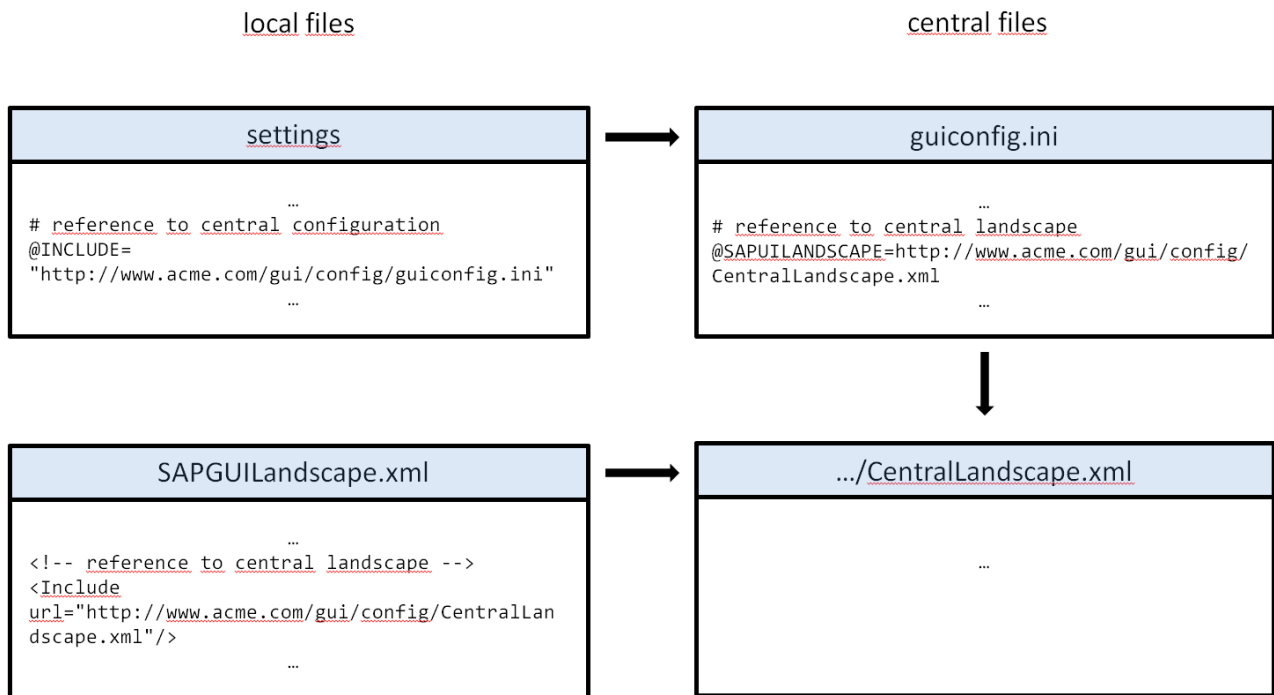
- Windows: <home directory>\AppData\Local\Low\SAPGUI\
- OS X: <home directory>/Library/Preferences/SAP/
- Linux: <home directory>/.SAPGUI/

Information stored in the *SAPGUI_Landscape.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 the SAP GUI for Java is started the first time, or they import the values stored in *connections* and *connectionTree.XML* from versions before 7.40, if present. Please note that this import only happens once.

1.1.3 Configuration Landscape Overview



1.2 Configuration Data Migration

- **One-Time-Migration:** If SAP GUI for Java has been used already, the data will be migrated automatically into the new landscape files with the first start of SAP GUI for Java 7.40. Only connections stored in the local files *connections* and *connectionTree.XML* are migrated. If no central SAP UI Landscape is configured, the migration process on the client creates locally stored message server and router entries from centrally managed for SAP GUI for Java 7.30 configuration to be able to connect to those systems. In the later case the local overwrites can be removed again later after a central SAP UI Landscape is configured (SAP GUI for Java 7.40: Preferences > Configuration: Logon > SAP UI Landscape: Remove Local Overwrites).
- **Continuous Update:** If you want to use SAP GUI for Windows or Java 7.30 and 7.40 in parallel, you do not have to maintain the configuration files separately. You can maintain the configuration data in either version and convert the configuration files then for the use in the other version. The conversion procedure is described under [Configuration Data Conversion](#). For detailed information on the new landscape format, see [SAP UI Landscape Format XML Description](#).

2 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.

2.1 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="2014-11-12T16:04:24Z" origin="http://acme.com/config/SAPUILandscape.xml"
localization="en">
- <Configuration>
  <Parameter value="http://acme.com/systemstatus" key="SystemStatusURL"/>
</Configuration>
- <Includes>
  <Include url="http://acme.com/config/SAPUILandscapeIT.xml"/>
</Includes>
- <Services>
- <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
  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">
- <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="3d6fe003-61f8-4510-82c6-e0d20bbabc22" serviceid="e1ab714e-50da-4cba-9454-
  5816d332d389"/>
  <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>
- <LDAPS>
  <LDAP name="LDAP" uuid="af6ffdec-6273-4e59-8a14-1f21bb0cf3e4" description="LDAP for ACME Corp."/>
</LDAPS>
</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*.

Note:

You 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 favorite list or 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, NWBC connection to WebAS ABAP or Portal, or search provider used by NWBC). 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.

General notes:

You can insert regular XML comments.

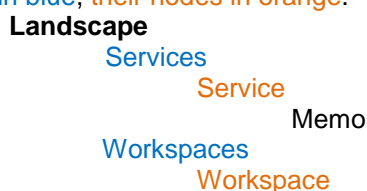
If a section remains empty, for example the `<Routers>` section, it has not effect.

2.2 SAP UI Landscape Format Specification

This section contains the detailed description of the XML format.

2.2.1 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**.



Item
 Node
 Messageservers
 Messageserver
 Routers
 Router
 Configuration
 Parameter
 INCLUDES
 INCLUDE

2.2.2 Special Attribute Types

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

Attribute	Description
<i>uuid</i>	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". Example: "d5bf6876-0ee9-4ae2-8c68-9aeb07081a5e"
<i>boolean</i>	A boolean value with "0"=false or "1"=true, if not otherwise specified.
<i>long</i>	Used for timestamp: number of seconds since January 1, 1970.
<i>service type</i>	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 .

2.2.3 Header Attributes

The root node of the xml is a *Landscape* tag:

Attribute	Description
<i>version</i>	Version of the landscape format; currently: 1
<i>origin</i>	Original location (source) of the current landscape as URL.
<i>updated [optional]</i>	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.

Note:

The data type of the attributes is string, if not specified otherwise.

2.2.4 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.

2.2.4.1 General Attributes

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

Attribute	Description
uuid [mandatory]	Unique ID of the service, to be referred to later for unique identification during merge with user (client-defined) services.
name [mandatory]	Name of the service to be displayed in service list
type [mandatory]	Service type attribute (see list of allowed type IDs below)
description [optional]	If not provided, name is used as default value.
memo [optional]	<p>If the description attribute is not sufficient, every service type can optionally have a <i>Memo</i> 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 <code>xml:space="preserve"</code>.</p> <pre><Service type="SAPGUI" uuid="d5bf6876-0ee9-4ae2-8c68-9aeb07081a5e" name="B30"...> <Memo xml:space="preserve"> Free text..... </Memo></pre>

2.2.4.2 Service-Type Specific Attributes

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

2.2.4.2.1 SAP GUI

In the following list, you find the additional attributes of the SAP GUI service type and their possible values.

Area	Attribute	Description
Host	mode	Mode for host specification. Possible values: <ul style="list-style-type: none"> 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"). 1 – application server. For direct application server mode (see the old "/H/applicationserverhost/S/port"). 2 – GUIParam (GUI Parameters). For Windows GUI legacy mode with "guiparam".
	msid [optional, uuid]	Message server ID from the message server repository.
	server [optional]	Fully qualified application server address (host + port) or logon group depending on the <i>mode</i> value. Examples: "abcd.acme.com:3206", "PUBLIC".
	routerid [optional, uuid]	Router ID from the router ID repository
SNC	sncop [optional, int]	SNC operation Possible values: 0, -1, 1, 2, 3, 9. 0 – SNC is disabled.
	sncname [optional]	SNC Name Example: "p/secude:CN=ACM, O=ACME, C=DE".
	sncnosso [optional, boolean]	No SSO, secure connection only
Encoding	encoding [optional]	Protocol encoding. Values are: "diag", "cdiag", "R3xml"
	protocol [optional]	Transport protocol. Values are: "ni", "R3http",
Content Encoding	sapcpag [optional, int]	SAP Codepage Examples: 0, 1, 2, 1100 etc.

	<i>uncoff</i> [optional, boolean]	Flag to disable Unicode. Backward compatibility.
	<i>cencoding</i> [optional]	Encoding. Backward compatibility. Examples: "UTF-8", "ISO-8859-1"
	<i>clocale</i> [optional]	Locale. Backward compatibility. Examples: "de", "en", "en_us", etc.
Upload/ Download	<i>dcpq</i> [optional]	SAP Code page for upload/download Examples: 0, 1, 2, 1100 etc.
Logon	<i>client</i> [optional]	Default client Example: "000", "100"
	<i>user</i> [optional]	Default user name
	<i>language</i> [optional]	Logon language
	<i>cmd_type</i> [optional, enum]	Command type Values: "Transaction", "Report", "SystemCommand".
	<i>cmd</i> [optional]	Command; content depends on <i>cmd_type</i> .
Shortcut	<i>shortcut</i> [optional, boolean]	Shortcut connection
	<i>reuse</i> [optional, boolean]	Reuse connection
	<i>connid</i> [optional]	ConnectionID
	<i>winmax</i> [optional, boolean]	Window maximized
	<i>systemid</i> [optional]	System name = SAP Database Name = SAP System ID <- for shortcut connection reuse. Example: "B30".
	<i>guiparam</i> [optional]	Legacy GUI command line parameters; <i>shortcut</i> should be set to 1, mode is 2 for GUIParam.
Misc.	<i>wan</i> [optional, boolean]	WAN flag
	<i>wp</i> [optional, boolean]	Workspace flag
	<i>rfcid</i> [optional]	RFC ID reserved for dialog RFC usage
	<i>sso2</i> [optional]	SSO token reserved for dialog RFC usage

2.2.4.2.2 NWBC

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

Attribute	Description
<i>url</i> [mandatory]	Service URL
<i>client</i> [optional]	Default client
<i>user</i> [optional]	Default user name
<i>language</i> [optional]	Logon language
<i>Msid</i>	Message server ID from the message servers repository

[optional, UUID]	
server [optional]	Fully qualified application server address (host + port) or logon group depending on the <i>mode</i> value Examples: "abcd.acme.com:3206", "PUBLIC".
slc [optional, boolean]	Usage of SLC (secure login client – NW SSO) for login
systemid [optional]	System name = SAP Database Name = SAP System ID Example: "B30".
sapguiid [optional, UUID]	Service ID from service repository for service of type SAPGUI. Used only to get a name and description of an existing SAPGUI connection.

2.2.4.2.3 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 NWBC.

Attribute	Description
url [mandatory]	Service URL
user [optional]	Default user name
language [optional]	Logon language
slc [optional, boolean]	Usage of SLC (secure login client – NW SSO) for login

2.2.4.2.4 MAP

In the following list, you find the additional attributes of the MAP service type and their values. The MAP service type is used by NWBC.

Attribute	Description
url [mandatory]	Service URL
user [optional]	Default user name
language [optional]	Logon language

2.2.4.2.5 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 NWBC.

Attribute	Description
url [mandatory]	URL string

2.2.4.2.6 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 NWBC. 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 [mandatory]	Service URL (<i>http://www.google.com</i>)
parameters [optional]	Parameters string with place holders for search terms (for example “?q={0}”, where {0} is placeholder for user search input)
mnemonic [optional]	Shorthand symbol that can be selected as search location in Quick Launch.

2.2.4.2.7 Reference

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

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

2.2.5 Workspaces 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.

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

Area	Attribute	Description
Workspaces (Contains a list of workspaces)	default optional	The « <i>Workspaces</i> » section contains a list of workspaces. You can set a default workspace: <i>default [optional]</i> – attribute specifying the default workspace ID from within the workspaces list. If not specified, the first workspace is used as default. If <i>Workspaces</i> section is empty, a default “unnamed” workspace will be assumed, which contains all services from the service list.
	uuid [obligatory, UUID]	Obligatory ID of the workspace
Workspace	name [obligatory]	Name of the workspace
	description [optional]	Description of the workspace
	timestamp [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 [optional]	Marks workspace as hidden and not visible in UI.
Node (Folder)	uuid [obligatory, uuid]	Obligatory ID of the node
	name [obligatory]	Name of the folder
	description [optional]	Description of the folder
	expanded [optional]	State of folder in UI: collapsed or expanded. Default value is ‘true’ (1=expanded).
	timestamp [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 [optional]	Mark node as hidden and not visible in UI. Allows changing workspace configuration locally.
Item (Service Link)	uuid [obligatory, UUID]	Obligatory unique ID of the item
	service [obligatory]	ID of the service this item refers to (should be one of known services from Services repository)
	timestamp [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 [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.

2.2.6 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 [mandatory]	Obligatory ID of the message server
name [mandatory]	Name of the message server, usually the system ID of the server.
host [mandatory]	Host of the server For example: “binmain.acme.com”.
port [mandatory, int]	Port number to message server service Example: “3276”.

description [optional]	Description of the message server
routerid [optional, uuid]	UUID of the router from Routers repository

2.2.7 Routers Repository

The *Routers* repository (collection node on 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 (see above). 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 [mandatory, uuid]	Obligatory id of the router
name [mandatory]	Name of the router
router [mandatory]	Complete router string Example: "/H/router01.acme.com/S/6756/P/abc123".
port [mandatory, int]	Port number to message server service Example: 3276.
description [optional]	Description of the router

2.2.8 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 [mandatory]	URL string to a landscape source
index [mandatory, integer]	Index to import, from lowest to highest
name [optional]	Name of the include
description [optional]	Description of the include

2.2.9 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 [mandatory]	Key for the parameter
value [mandatory]	Value of the parameter
timestamp [optional, long]	Attribute used on client side to note the time stamp of the change applied to the item by the user. Based on the time stamp, the client application can decide when to purge dangling user modifications, when item is deleted on server side.

3 CONFIGURATION DATA CONVERSION

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.

3.1 Prerequisites

- installed SAP GUI for Java 7.40

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)

3.2 Convert File Creation

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

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 $@
```

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 $@
```

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 %*
```

3.3 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. 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:

a. 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:

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

To **convert SAP GUI for Windows 7.30** *connection data*:

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

b. 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:

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

To **convert SAP GUI for Java 7.30** *connection data*:

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

4. 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:

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

To **convert SAP GUI for Java 7.40** *connections*:

```
converter exportConJ SourceLandscape connectionsFile connectionTreeFile
```

4 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 the SAP GUI for Java Manual 5.3.3 *Trace Information*.



© 2016 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.
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. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. 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 SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE 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.
In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.