Winshuttle and SSO

SSO Methods Supported by Winshuttle Applications

WINSHUTTLE™

SAP Partner

Microsoft Partner

Contents

SSO – An Introduction

Single Sign-On (SSO) is a specialized form of software authentication that enables a user to authenticate once and gain access to the resources of multiple back-end software systems. SSO enables authorized users to reliably and transparently access software resources across technical system boundaries.[1]

With the convergence of business applications toward enterprise based solutions, security models need to be homogenized to allow for a consistent authentication policy. SSO users would generally authenticate themselves over a single authentication authority and would gain access over all the resources that have established a trust connection with the authentication authority.

This Winshuttle paper aims at discussing the SSO methods that Winshuttle applications work with and the process flows when working with SSO together with SAP systems and Winshuttle products.

Winshuttle product use in environments without SSO

Winshuttle products communicate with SAP using standard available Remote Function Calls (RFCs). A Winshuttle application sends the required connection parameters to the RFC interface which is then responsible for making any further modifications to the parameters and hence establishing a connection with a given SAP instance. There are different logon methods available with Winshuttle products when communicating with SAP which the following section will discusses in further detail.

Saplogon.ini based logon

A typical installation of the SAP GUI uses a local GUI configuration file to store all the configuration information of the SAP instances that the GUI client communicates with. The ini file stores all connection parameters for SAP systems defined within the SAP GUI such as IP address, logon groups, message servers, IP addresses etc. The default location of the configuration file (saplogon.ini) depends on the platform on which the SAP GUI is installed. However, this can be configured according to the specific requirements of a given environment. Saplogon.ini forms a critical component for most logon mechanisms used within Winshuttle applications. The connection to a selected SAP system is achieved by the means of a connection string that is passed to the system by the application's logon component. For a basic username/password based logon, a user selects the SAP system he wants to logon to, with the logon credentials. The list of the SAP systems is populated on the basis of the saplogon.ini file. Winshuttle respects the SAP GUI reading order for locating the configuration file. The location of this file can also be set from within the Application Options -> SAP defaults in Winshuttle products. This setting becomes useful in the context of multiple saplogon.ini files or saplogon.ini files located on network drives mapped to the local machine.

In case a machine's SAP system configuration does not include a saplogon.ini file, the application fails to load the SAP system information. In this case, the user would be required to provide the system details manually. This information would include attributes such as application server host, system number etc. manually under server tab on the logon dialog. The details would be directly used to create the connection string for further logon.

1 http://help.sap.com/saphelp_nwpi71/helpdata/en/44/0ebf6c9b2b0d1ae10000000a114a6b/frameset.htm

The connection string follows a standard format which is described below along with the individual elements. This connection string is used to create a connection to the SAP system using the librfc32.dll that typically comes as part of the SAP GUI installation. We will be using this format to illustrate how connection parameters are created and passed. Based on the logon type selected by the user, the underlined parameter will change. The remainder of the connecting string is the same and is populated with appropriate data supplied by the user.

Table 1: Saplogon.ini based logon

CLIENT=<Client_Id> USER=<Uname> PASSWD=****** LANG=<Lang> **SAPLOGON_ID=[SYS_DESC_In_IniFile]**	**CLIENT**	Client Number
	USER	User Name
	PASSWD	Password
	LANG	Logon Language
	SAPLOGON_ID	Selected system description used to search saplogon.ini

In situations where the saplogon.ini file is not present, the user can choose to provide the details by using the server tab. Instead of SAPLOGON_ID, the actual system connection details are added through this method. These details would in turn depend on whether the system is an application or a message server.

Table 2: Application and Message server logon

Application server logon		
CLIENT=<Client> USER=<Uname> PASSWD=****** LANG=<Lang> **ASHOST=<Application Server Host> SYSNR=<system number>**	**CLIENT**	Client Number
	USER	User Name
	PASSWD	Password
	LANG	Logon Language
	ASHOST	Application server host
	SYSNR	System Number
Message server logon		

CLIENT=<Client> USER=<Uname> PASSWD=****** LANG=<Lang> **GROUP=<GrpName> MSHOST=<Message server host>**	**CLIENT**	Client Number
	USER	User Name
	PASSWD	Password
	MSHOST	Message server host
	GROUP	Group name

Unattended Logon with the Winshuttle ALF and Windows scheduler

ALF or Auto logon files enables automatic logon to SAP systems without the need to provide logon details. This is different from SSO logon as it is, in essence, a way to store the credentials in an encrypted file on the local machine. Instead of manually entering the details, this file is used to retrieve the connection string to connect to SAP. To avoid any exposure of the SAP credentials, the contents of the ALF are encrypted using an AES-128[2] encryption algorithm.

To create an ALF, a user will capture logon details from the logon dialog box in the Winshuttle application. The user is required to only select the SAP system and provide the logon credentials, as in a normal logon step. The Winshuttle application will automatically retrieve the system connection details from saplogon.ini and then the captured details would create a connection string similar to the ones discussed in Table 2. The role of saplogon.ini comes in only during the capture step where the system details are populated based on user selection. For a configuration without the saplogon.ini, the details can be provided directly. The connection string is stored in an encrypted format in the auto logon file (ALF). When the user runs a script that has an associated ALF, saplogon.ini file is not required and the connection string from the file is used to create a connection with SAP.

For a scheduler, the user is required to capture the SAP details. These details are stored with the scheduled job and at the time of scheduled run, the connection string parameter will be used to connect to SAP and the ALF is used to pass the credentials at run-time.

Logon with Central and Server

Winshuttle's enterprise application, Winshuttle Central, provides another option - the capability of an automated way of connecting to an SAP system. The process is similar to the one used for creating ALFs with a provision to save this on the Central site. At the time of logon, based on the user's Central credentials and the SAP system selected, the relevant connection string is returned to the application. The application, in turn, uses this to create a connection with SAP. This implies a logon without any manual intervention, other than selecting the required SAP system.

If a script is run on Winshuttle Server, as in the case of auto post, the Central credentials are passed to Server based on the SAP system the file is designated to run with, and the Server creates a connection with SAP.

For InfoPath, Adobe or any other forms created using Winshuttle web services, the user can pass the SAP credentials as a part of the form, or an appropriate connection string could be selected by Winshuttle Server based on the windows domain identity of the domain authenticated submitting user.
Currently, Central and Server do not support Single sign on methods for logon to SAP.

Winshuttle products and Single sign on

Winshuttle products allow users to leverage Single sign on (SSO) capabilities. SSO is available for SAP GUI for Windows as well as with SAP Portal. SAP logon for 'SAP GUI for windows' is termed SSO1, while for the Portal SSO2. These logon mechanisms are available for normal logon as well as logon through ALF and scheduler. The ability to logon via the SAP portal does however still require the presence of the SAP GUI or at the very least the LIBRFC32.dll. Winshuttle products do not communicate with the SAP Portal for anything other than authentication.

2 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Out of the commonly available ways of implementing SSO, Winshuttle solutions work with the following methods

- SSO for SAP GUI (SSO1)
  - NTLM
  - Kerberos
- SSO for Web based Application (SAP Portal, SSO2)
  - SAP logon ticket

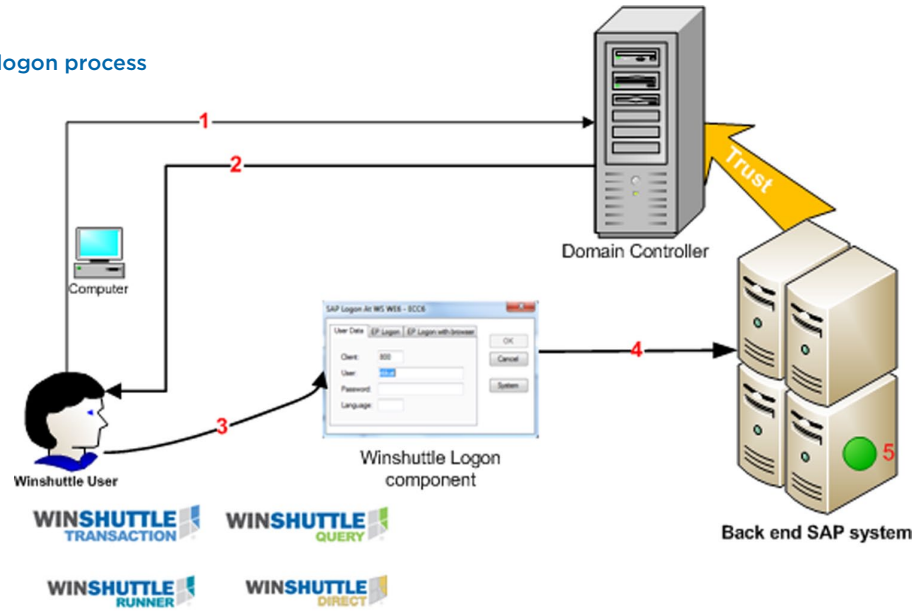
SSO1

Winshuttle works with SSO1 implemented using Microsoft Windows NTLM or Microsoft Kerberos. This is typically contained in Microsoft operating systems. This method uses an Integrated Windows Authentication method and is delegated to the Windows platform with NTLM or Kerberos based on the presence of a Windows Domain Controller. SAP provides the Secure Network Communication (SNC) option to provide end-to-end support for SSO. SNC aims to integrate external security products with SAP systems in a pluggable fashion, providing authentication as well as data protection at the same time. For authenticating users, SNC utilizes an external product to delegate authorization tasks. Based on the external authentication result, SNC can grant users the corresponding access to the backend SAP systems using a secure network communication method that is more secure than the standard http communication method. SNC secures the communication paths between the SAP GUI as the front-end and the back-end SAP system.

Creating connections to SAP

The user (1) initially logs on to the windows machine and (2) is authenticated by the Active Directory. The user now selects the SSO enabled SAP system and (3) logs in without providing the username/password information. (4) The Winshuttle Logon component sends an RFC logon request to the SAP system. The information regarding whether the SAP system is SNC enabled and the corresponding details are fetched from the saplogon.ini file. The Winshuttle logon component sends this information to the back-end SAP system. (5) At the time of logon, the SAP system recognizes that this is an SNC enabled logon and authenticates the user based on the trust relation set up with the domain controller.

Figure 1: SSO1 logon process



Saplogon.ini based logon

The user selected SAP system details are retrieved from the saplogon.ini file for connection properties. The connection string is as depicted in Table 1. The password details are provided as dummy values. The actual authentication is done based on SSO setup used by the organization.

App server logon for ALF and Scheduler

The application server logon in this case only differs in the fact that the SAP system connection properties need to be specified. Users are not required to explicitly enter these values; they are instead 'captured' from within the application. The user is required to only select the SAP system and provide the logon credentials, like a normal logon step. The Winshuttle application will automatically retrieve the system connection details from saplogon.ini. The logon component also sends in properties like SNC mode and SNC partner name to clarify that this is an SSO logon.

Message server logon for ALF and Scheduler

For message server logon as well, the connection details corresponding to the message server group and host need to be provided by the logon component in addition to SNC related information.

Table 3: Application and Message server logon for SSO1

Application server logon		
CLIENT=800 USER=username1 PASSWD=****** LANG=EN ASHOST=<Application server> SYSNR=<system number> SNC_MODE=true SNC_PARTNERNAME=<partner>	**CLIENT**	Client Number
	USER	User name
	PASSWD	Password (dummy in this case)
	LANG	Logon Language
	ASHOST	Application server host
	SYSNR	System number
	SNC Mode	SNC Mode: true/false
	SNC Partner	Partner name
Message server logon		
CLIENT=800 USER=username1 PASSWD=****** LANG=EN GROUP=GrpName MSHOST=<Message server> SNC_MODE=true SNC_PARTNERNAME=<partner>	**CLIENT**	Client Number
	USER	User name
	PASSWD	Password (dummy in this case)
	LANG	Logon Language
	MSHOST	Message server host
	GROUP	Group name
	SNC Mode	SNC Mode: true/false
	SNC Partner	Partner name

SSO2

In this Web-based scenario within the SAP context, SAP Netweaver AS is involved – either as the authenticating authority or as a resource server. In some instances it is used for initial authentication while in others for both initial authentication and SSO. The initial step of user authentication aims at ensuring user identification. The user has to provide his credentials in the form of user ID/password to the authentication authority. The authentication authority verifies the submitted identity and grants that user access to the back end systems.

Winshuttle applications support SSO2 by way of three different mechanisms.

1. Enterprise Portal logon using Username/Password
2. Enterprise Portal logon using SPNego
3. Enterprise Portal logon with browser

The first two methods (username/password and SPNego) are independent of the browser in use, but will require ports to be configured on the client machine to be able to send an http request to the portal for the logon ticket. In the third method, EP logon with browser, there may be a dependency on the browser installed on the machine. It is recommended to use Microsoft Internet Explorer 6.0

SAP Enterprise Portal logon with Username/Password

In this method, (1) the user provides the initial username password and the portal address. (2) Winshuttle logon component sends a web request to the portal server. The portal server authenticates the user based on this username and password. (3) The ticket issuing authority issues a ticket in the form of non-persistent cookie by the name of "MYSAPSSO2" in the browser sessions. A Winshuttle logon component reads the ticket and adds system connection information. (4) The combined information is then passed to the back-end SAP system using an RFC logon. The user is then validated on the SAP system and can perform record/run actions.

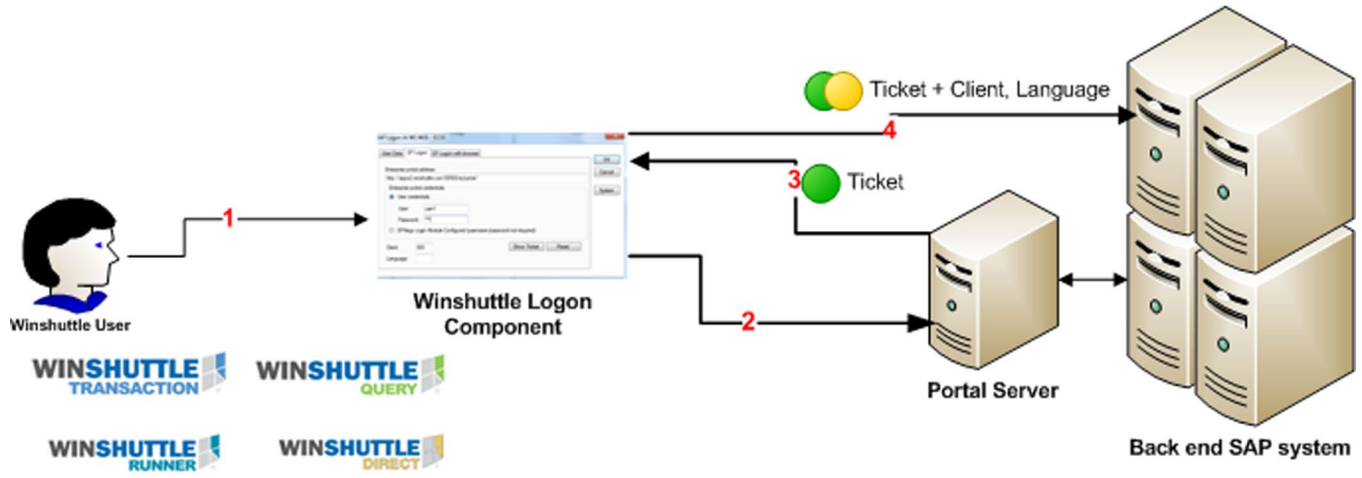


Figure 2: Username/Password based SSO2 process

In addition to the connection information to the SAP system (which corresponds to the SAP GUI), the SAP logon ticket is also passed as MYSAPSSO2 variable for saplogon.ini based logon.

The message server logon and the application server logon applicable for Scheduler and ALF remains the same with the difference in the connection part, as demonstrated in the earlier examples with SSO1 and normal RFC logon. The SAPLOGON_ID parameter would be replaced by the SAP system connection details instead.

Table 4: Saplogon.ini based logon for SSO2

CLIENT=800 USER=username1 PASSWD=****** LANG=EN SAPLOGON_ID=[SYS_DESC_In_IniFile] MYSAPSSO2=<Logon ticket>	**CLIENT**	Client Number
	USER	User Name (dummy)
	PASSWD	Password (dummy)
	LANG	Logon Language
	SAPLOGON_ID	Selected system description used to search saplogon.ini
	MYSAPSS02	SAP logon ticket in form of non persistent cookie

SAP Enterprise Logon with SPNego

SPNego is a platform independent way of implementing initial user authentication using an integrated windows authentication model. It is typically used when users are using Windows domain accounts in combination with Kerberos for logging on. Users do not need to apply any username or password for initial authentication other than their initial Windows Logon credentials at startup.

For Enterprise logon with Winshuttle applications, SPNego-based logon differs from the username/password-based logon only with respect to the initial authentication. The initial authentication is handled completely by the SPNego module. At the time of logon to the SAP system, (1) the user supplies the portal address and selects SPNego authentication. (2) As a part of the SPNego module, the user is authenticated by the Domain Controller that acts as ticket granting authority. The user is validated on the portal based on this token and (3) a ticket is returned to the browser sessions in form of a cookie. Winshuttle logon component reads this information and (4) and appends other connection information (see Table 4) and sends it to the back end SAP system in form of an RFC logon request.

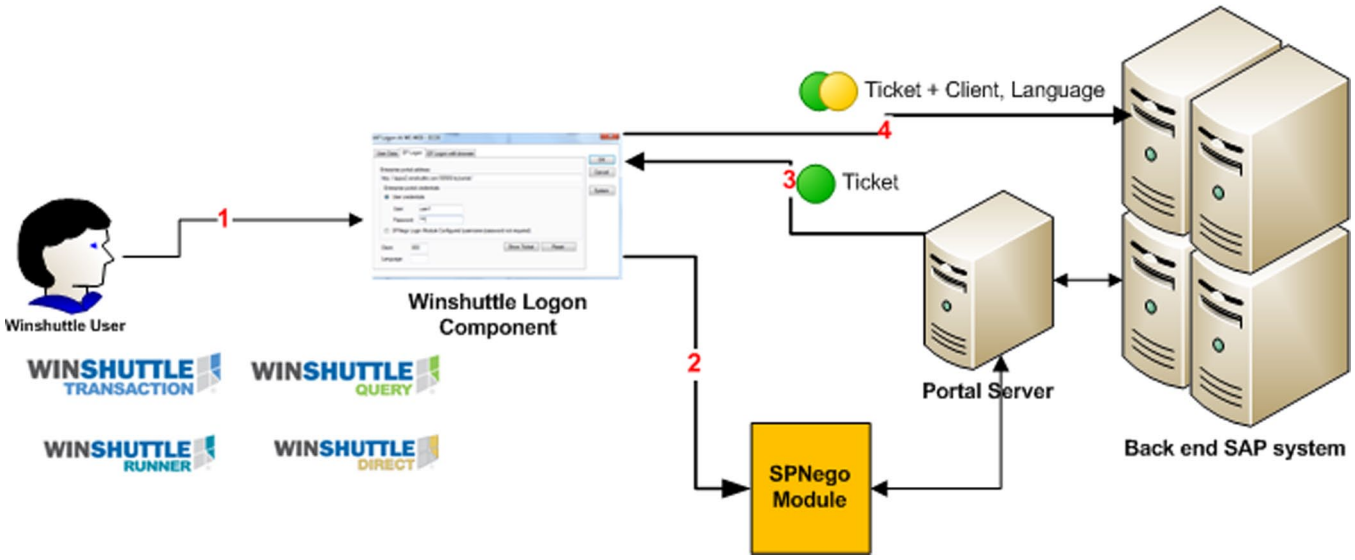


Figure 3: SPNego based SSO2 process

The SPNego module initial authentication can be explained from the following steps:

1. An HTTP get request is sent to the Portal server
2. An access denied request informs the browser that it is dependent on Kerberos
3. The browser then sends the request to Domain Controller
4. The user is authenticated and a ticket is issued within SPNego token
5. The ticket wrapped in the token is sent to the Portal
6. The portal validates the token and authenticates the user

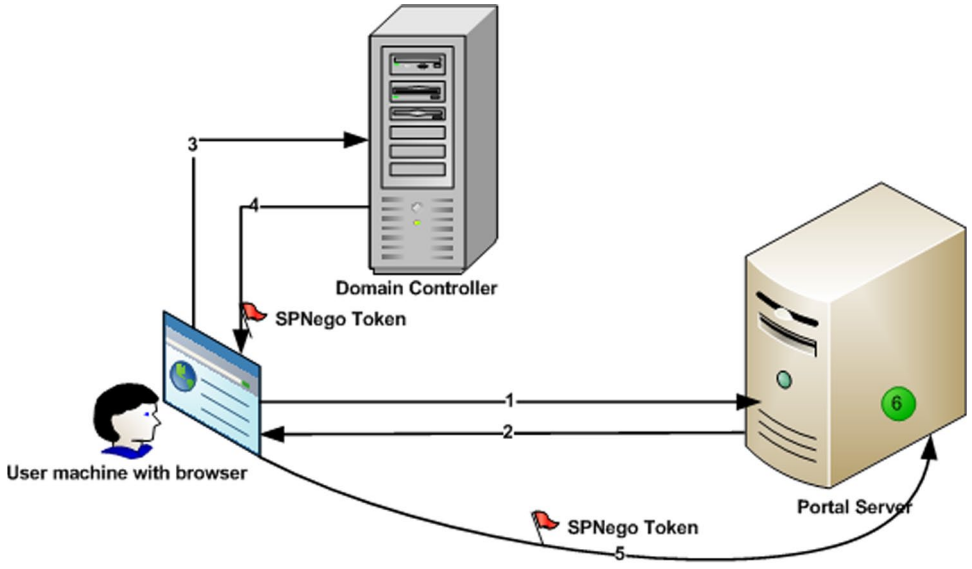


Figure 4: SPNego Authentication

SAP Enterprise Portal logon with browser

This method is very useful in the cases where the portal is unable to authenticate the users directly based on username/password and may need user intervention. The user can enter logon details, which may be more than just username/password depending on the portal configuration, in a browser available in the logon dialog. On the logon page, (1) the user provides the portal address to logon to. (2) After the user reaches the logon page, he can provide the user name, password and other details for authentication. (3) Once the user is authenticated on the browser, (4) the cookie containing the ticket information is returned to the browser and the logon component reads this information from the browser. (5) The Winshuttle logon component appends other connection information (see Table 4) to the ticket and makes a connection to the back-end SAP system using an RFC logon.

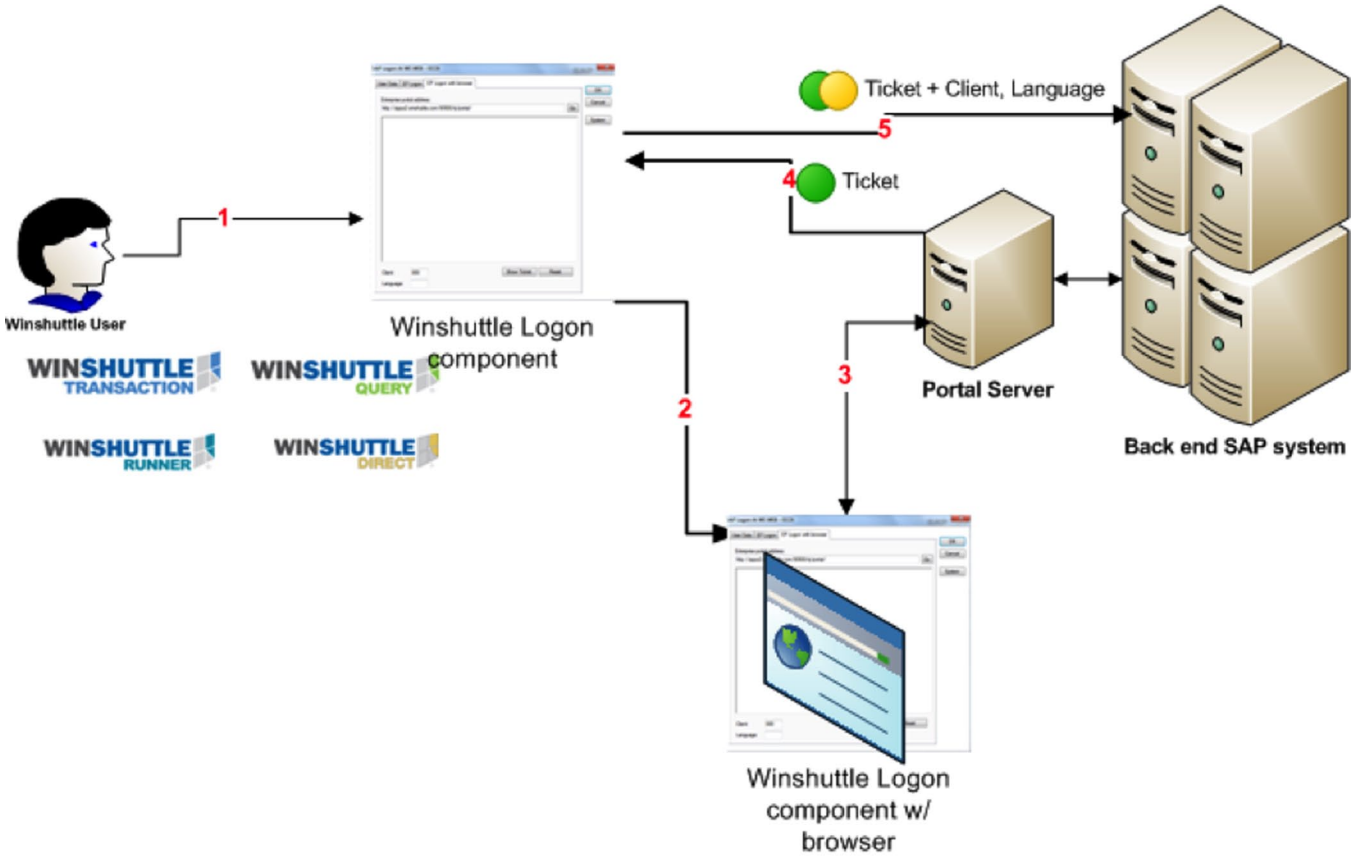


Figure 5: SSO2 with browser process

References

1. Tilo Boettcher, Juergen Daiberl, André Fischer, Lei Liu, Unleash the power of Single Sign-On with Microsoft and SAP, September 2007
2. http://help.sap.com/saphelp_nwpi71/helpdata/en/44/0ebf6c9b2b0d1ae10000000a114a6b/frameset.htm
3. http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/7b93dbf4-0901-0010-2e8d-bfacd5fdce81#q-2

Winshuttle is the ERP Usability Company, providing software products that enable business users to work with SAP directly from Excel, Web forms and other interfaces without any programming. Winshuttle focuses on a simple fact – when using SAP applications, time is money. Winshuttle's usability solutions radically accelerate SAP user transactions, saving and redirecting millions of dollars for SAP's customers every day. These financial benefits are achieved by significantly reducing employee and contractor costs and increasing resources to address more strategic priorities. Thousands of customers use Winshuttle to make their SAP lives easier.

Headquartered in Bothell, Washington, Winshuttle has offices in the United Kingdom, France, Germany, and India. For more information, visit www.winshuttle.com.

WINSHUTTLE™

Microsoft Partner

SAP Partner

Corporate Headquarters
Bothell, WA
Tel + 1 (800) 711-9798
Fax + 1 (425) 527-6666
www.winshuttle.com

United Kingdom
London, U.K.
Tel +44 (0) 208 704 4170
Fax +44 (0) 208 711 2665
www.winshuttle.co.uk

Germany
Bremerhaven, Germany
Tel +49 (0) 471 140840
Fax +49 (0)471 140849
www.winshuttle-software.de

France
Maisons-Alfort, France
Tel +33 (0) 148 937 171
Fax +33 (0) 143 683 768
www.winshuttle.fr

India
Research & Development
Chandigarh, India
Tel +91 (0) 172 465 5941
www.winshuttle.in