

ObserveIT Service Desk Integration Guide

Contents

1	Purpose of this Document	2
2	Overview and Architecture	2
3	Web Services Integration.....	3
4	Customizing a New Service Desk System	4
5	Appendix: Web Service Service Desk Parameters.....	5
5.1	Ticket Information	5
5.2	Agent Information	5

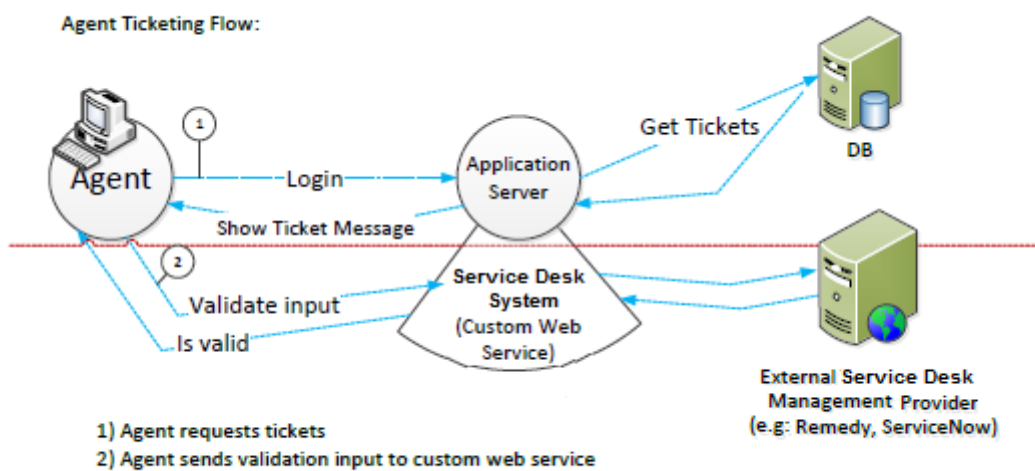
1 Purpose of this Document

The purpose of this document is to provide ObserveIT customers and partners with information that will facilitate the process of customizing a Service Desk system and integrating it with ObserveIT's session recording system. This document provides instructions on how to build a Web Service that will enable customers to implement the integration of ObserveIT with their own Service Desk system.

2 Overview and Architecture

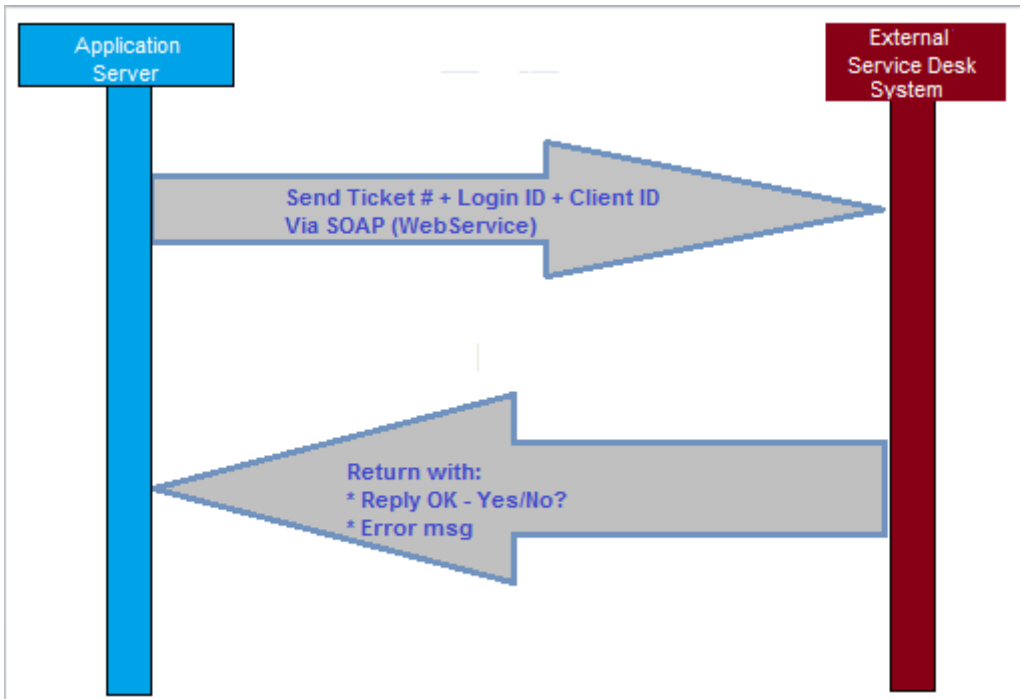
When ObserveIT's session recording system is integrated with a Service Desk system, selected IT administrators or remote vendors are required to enter a valid ticket number from a system, in order to complete the login process to a corporate server. The ticket number that is entered by the user must be validated against an external Service Desk system.

The following diagram illustrates how Agent ticketing information flows between the ObserveIT Agent and an external Service Desk system.



3 Web Services Integration

Web Services are used to communicate and pass relevant information between the ObserveIT Agent and the Service Desk system, via the ObserveIT Application Server, as illustrated in the following diagram.



The integration between ObserveIT and the Service Desk system is based on a WSDL (Web Services Description Language) XML-based document that contains a set of definitions to describe the Web Service. Web Services exchange information by using the SOAP XML-based protocol.

The Web Service can act as a gateway between the ObserveIT Agent and the Service Desk system, by transmitting information from the ObserveIT Agent (such as the ticket number and other information about the login session automatically provided by the Agent) to the Service Desk system, and synchronizing the relevant response information from the Service Desk system to the ObserveIT Agent.

The Web Service can also serve as the endpoint itself by providing the appropriate validation response to the input and request information from the ObserveIT Agent. In this case, the ticket validation is performed by the customized Web Service.

To facilitate the Web Services integration process for customers, the ObserveIT installation package includes a template project for a Web Service that was created by ObserveIT using WSDL. The functions in the WSDL template demonstrate how the customer Web Service should be built.

4 Customizing a New Service Desk System

In order to customize a new Service Desk system, you must first create your own Web Service, and then define this Web Service in the ObserveIT Web Management Console.

➤ To create your own Web Service:

1. Implement ObserveIT's Web Service Template Project for Visual Studio 2010 in your developer environment.

Note: The ObserveIT Web Service Template Project "ObserveIT.TicketingImplementation-x.x.x.zip" is included in your ObserveIT installation package under the "Docs" folder.

The Web Service should look like this:

```
{
  /// <summary>
  /// Summary description for Service1
  /// </summary>
  [WebService(Namespace = "http://observeit.com/validation/ticketing")]
  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
  [System.ComponentModel.ToolboxItem(false)]
  public class Service1 : TicketingTemplate
  {
    [WebMethod]
    public override IsTicketValidResponse IsTicketValid(CustomTicketMessage aTicket, AgentInfo aAgentInfo)
    {
      IsTicketValidResponse ret = new IsTicketValidResponse();
      ret.exists = aTicket.TicketNumber == "123";
      return ret;
    }

    [WebMethod]
    public override bool TestConnection(string aUsername, string aPassword)
    {
      return true;
    }

    [WebMethod]
    public override CreateNewTicketResponse CreateNewTicket(CustomTicketMessage aTicket, AgentInfo aAgentInfo)
    {
      CreateNewTicketResponse ret = new CreateNewTicketResponse();
      ret.isCreated = true;
      return ret;
    }
  }
}
```

In the above example, a request is sent to the Service Desk system to provide a ticket ID for the user to enter in the Ticket Window; the ticket is then validated by the system. Note the following:

- The Web Service receives request information from the ObserveIT Agent, such as, "IsTicketValid" or "CreateTicket".
 - The system returns a response via the Web Service, such as, "IsTicketValidResponse" or "CreateNewTicketResponse", depending on the requested operation.
2. **In the code, define your own Web Service implementation according to your requirements.** For a description of the parameters you may need to define, see [Appendix: Web Service Service Desk Parameters](#).

Important: After you have customized your Web Service, you must define the new Web Service in the ObserveIT Web Management Console and check that it connects successfully to the Service Desk system. Each Service Desk system must have a name and a URL that specifies the location of the Web Service. Please refer to the product documentation for detailed instructions about [Configuring Service Desk Systems](#).

5 Appendix: Web Service Service Desk Parameters

In order to define your Web Service, you may need to configure the following parameters:

5.1 Ticket Information

Parameter	Description
TicketNumber	Ticket number that the user entered in the Ticket Window for validation.
SessionId	Unique ID of the current ObserveIT session.
Comments	Comments that the user entered in the Ticket Window.
IsReplyMandatory	Is the user required to send a text reply to the ticket message via a comment in the Ticket Window?
SlideViewerURL	URL to the playback of the current session.
IsBreakGlass	Is the user allowed to access the system without a valid ticket number?
ValidationCode	<i>Not currently used by the Web Service. For future implementations only.</i>
IsValidateUserID	<i>Not currently used by the Web Service. For future implementations only.</i>
IsValidateServerID	<i>Not currently used by the Web Service. For future implementations only.</i>

5.2 Agent Information

Parameter	Description
ServerId	IP address of the server on which the Agent is installed.
Login	Logged-in user name (Windows Authentication only).
Domain	Domain name of the Agent computer.
ServerName	Name of the server on which the Agent is installed.
SessionId	Unique ID of the current ObserveIT session.
ClientIP	IP address of the remote client.
ClientName	Remote client host name.
User	Logged-in user name (ObserveIT secondary authentication).
UserDomain	Logged-in user domain name (ObserveIT secondary authentication).