

Wiki

Article - Employ SAP GUI for HTML instead of an Internet service

Added by [Thorsten Domsalla](#) , last edited by [Jyothi](#) on Oct 01, 2008 ([view change](#)) [SHOW COMMENT](#)

Labels: [its_howto](#), [its](#), [internet_transaction_server](#)

Often, Internet Application Components (IAC) are used for particular services. Though, in many cases the SAP GUI for HTML provides more flexibility, a better user interface and session management. In the following we want to show a way to replace an IAC service with a modified SAP GUI for HTML service with restrictions in user navigation.

Target audience

Application architects, developers and technical consultants who want to provide a full-featured, HTML based user interface for Internet or Intranet services based on a ABAP applications or dynpros without the need to implement HTML Business templates or Javascript code.

Introduction

At first, I like to illuminate the concepts of IACs, EWTs and the SAP GUI for HTML.

IAC, EWT, SAP GUI for HTML or WebGUI?

Simply spoken, an IAC is a service running on an ITS that is not an SAP GUI for HTML or an EWT service. An SAP GUI for HTML service is a service with the parameter `~webgui` set to '1'. Often the term *WebGUI* is still used instead of *SAP GUI for HTML*. The SAP GUI for HTML supports all SAP GUI functionality (except for a small number of non-HTML compliant applications like the Graphics tool). An *EWT* is a service with the parameters `~transactiontype` set to 'EWT' (and also no `~webgui` set). Often, IAC or EWT services are denoted as *Internet services*.

Propably you also are familiar with another service parameter of the ITS: `~generatedynpro`. This parameter can be used in an IAC or EWT to allow a service to work without a given HTML Business *template*. A template contains instructions how to generate and render a particular page for a dynpro. If no template exist for a screen of a dynpro the ITS displays an error message. However, it is possible to let the ITS do the job and generate an appropriate HTML document for a page. This is what the parameter `~generatedynpro` is for. Though, there is a downside using the parameter `~generatedynpro`.

Among other disadvantages on using `~generatedynpro` the ITS can get confused about the way it must render a page. A page automatically generated with `~generatedynpro` is, so to speak, neither fish nor fowl. The ITS needs to make several considerations about the application and the Internet service as well as about the user interface. It really does a pretty good job in doing it, though there are still situations where the result might not be acceptable.

Another confusion exists about the usage of ERP controls, e. g. the ALV Grid. Those controls are supposed to work in an SAP GUI (for HTML) environment and support a lot of sophisticated features that are either not allowed or undesirable in Internet services. Some application developers try to overcome the restrictions in implementing a complex control in an IAC by using `~generatedynpro`. At first glance it seems to work this way, though in more complex scenarios it might fail due to unforeseeable side effects or simply the limited functionality of the IAC environment.

Conclusion: The use of `~generatedynpro` in IAC or EWT services **is not recommended!**

Summary: An SAP GUI for HTML service is specified by the parameter `~webgui` in the service definition. An EWT is a service primarily is based on templates for particular dynpros. An IAC service is neither an SAP GUI for HTML nor an EWT service and intended for Internet driven software.

Using SAP GUI for HTML instead of an IAC or EWT service

As we have learned earlier in this article the SAP GUI for HTML provides a full-featured user interface with full support of ERP controls and a sophisticated client-side session management. The question is

Why did we not use an SAP GUI for HTML instead of an IAC or EWT service before?

Although, or better: because the SAP GUI for HTML supports all features of an SAP GUI many ITS driven applications are implemented as IAC or EWT services. Some of the reasons follow:

1. The SAP GUI for HTML does not allow to use generated styles
2. A developer can provide a custom visualization of end user interface for an application through HTML Business templates and custom Javascript code
3. The user interface of the SAP GUI for HTML is much too complex or just too special for an Internet service
4. Through navigation elements like the Menu, the Back button or the OK Code Field an user might be able to navigate to the entire ERP system
5. No title should be shown

However, many applications mainly in the Intranet do not need restrictions as of 1.) or 2.). More than that, Intranet applications are supposed to support a fully featured user interface as provided by the SAP GUI for HTML because they want to make use of the complex ERP controls, context menus, function keys and so on. At least only points 4.) and 5.) are left as argument for the implementation of an IAC or EWT service.

Times and SAP GUI for HTML are changing

With notes **1010519**, "SAP GUI for HTML: Simplified Title Area Without Menu and OK Code" and **959417**, "Integrated ITS, Closing Session When Transaction Ends" the SAP GUI for HTML introduced two new service parameters, that will allow the developer to change the SAP GUI for HTML's appearance and behaviour: The appearance of the top *custom user area* (CUA) including the title bar, the menu and the applications button bar is configurable. Also, the navigation to other transactions or programs is restricted in a way that only the transaction defined in the service can be executed. Thus, the developer is able to define a service running the SAP GUI for HTML while not allowing the user to navigate out of the box.

~singletransaction

Often in a service driven environment the user is allowed only to call and work in particular services or transactions, e. g. in a service running inside a Portal's iview. With the user navigating from the current transaction of program the service is supposed to be closed. Of course the user must be able to follow program execution paths that are defined by the particular application called by the service.

Such a requirement is met by the ITS service parameter ~singletransaction. It is hooked deeply in the ITS server and backend server to look for any jump out of the current program's execution path or call stack. Whenever the user leaves the transaction that was called with the service start the session is logged off.

~webgui_simple_toolbar

Whereas Internet services only show a simple area with a title and a button to exit the service the SAP GUI for HTML features a full custom user area (CUA) at the top of the window with a title bar, menu, system buttons and so on. Because of that, from the perspective of a service provider, running the SAP GUI for HTML as a service is sort of overkill. That is why many ITS based services are implemented as IAC or EWT services.

Actually, the service parameter ~webgui_simple_toolbar was introduced to provide a solution for such a requirement. It allows to control the appearance of the CUA in lot of variations, but in a simple manner.

It can control the display of

- the title bar
- additional buttons "Cancel" and "Help" (like in EWT services)

- the system menu
- the tools buttons
- the application button bar
- and finally the information tab in the statusbar (new)

and any combination of all of those parts. **The OK code field is omitted in any case.**

Think of the parameter as sort of bitmap where a particular bit describes a particular part of the CUA. So the bit

1. turns on the **title bar**
2. turns on the **button "Cancel" and "Help"** (as in the EWT services)
3. turns on the **tools buttons** (e. g. "Back", "Print")
4. turns on the **system menu**
5. turns on the **application button bar.**

Using bit 7 you can turn off the information tab in the status bar (the area on the right side of the status bar with information about the server et al.)



Bit 0 (zero) has a special meaning. As of this writing it provides a recommended default setting with bits 1, 2 and 5 turned on for user's (developer's) convenience. Thus, it shows the title bar, the "Cancel" and "Help" buttons as well as the application button bar.

To find the correct value for the parameter `~webgui_simple_toolbar` just calculate the sum of 2 to the power of the bit position for any bit you want to set. For instance, if you want do show only the application button bar, you set `~webgui_simple_toolbar` to 32 (2 to the power of 5, in a short way 2^5). If you want to show additionally the title bar and also the "Cancel" and "Help" buttons you have to set the parameter `~webgui_simple_toolbar` to 38 ($2^1 + 2^2 + 2^5 = 2 + 4 + 32$) [however, since this is the default setting you could simply set `~webgui_simple_toolbar` to 1 (one) 😊].

Please note that if you set bit 1, it will override all other bits and show the page in the default manner with title bar, "Cancel" and "Help" button and the application button bar, e. g. if you set `~webgui_simple_toolbar` to 7 it will be handled exactly as if you set to 1.

Examples

Now let us have a look at some examples for different values set for `~webgui_simple_toolbar`:

Value	Screenshot
1	
32	

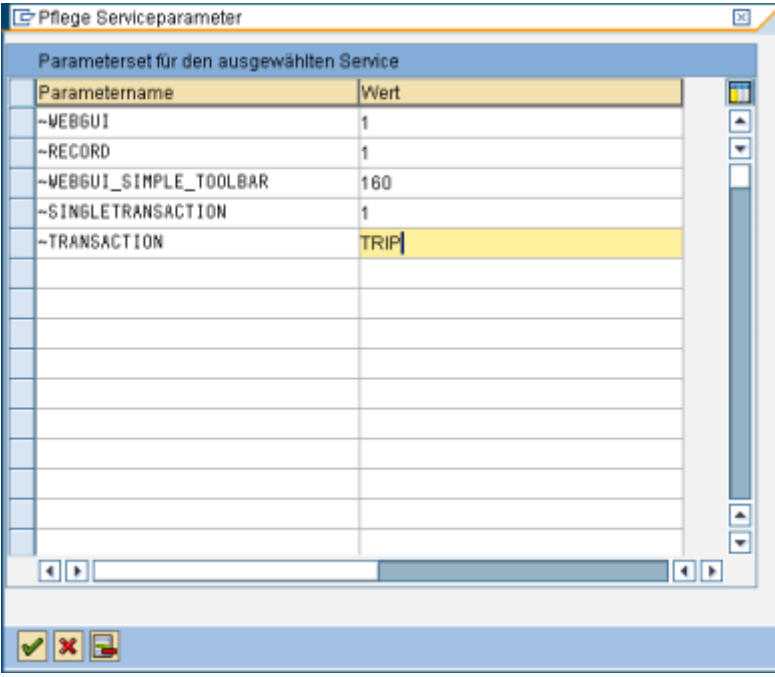


Putting it altogether

In the following we want to go in medias res and exemplarily create a service for the *Travel Manager* (transaction TRIP) using SAP GUI for HTML instead of an IAC or EWT service. Since the service is designed for running in a Portal, we do not want to display the title bar. Only the application button bar must be displayed. And, at least, we do not want to display any server information in the statusbar below the user area of the dynpro.

First Step

We copy the webgui service to a new service. In this example I copied /sap/bc/gui/sap/its/webgui to /sap/bc/gui/test/its/webgui_tj and added the parameters ~singletransaction, ~webgui_simple_toolbar and ~transaction.



We set ~singletransaction to make sure that the user can not leave this application and navigate into another program.

The parameter `~webgui_simple_toolbar` is set to value 160 ($2^5 + 2^7$) to display the application button bar only and omit the status bar information tab.

Second Step

Secondly ... there is no second step. 😊 We are finished with our task.

What we get when we start our new service looks like



As you can see, all GUI elements as popups, calendar control, HTMLviewer controls and all the others are supported as in SAP GUI for HTML - of course, since we *are* using the SAP GUI for HTML.

Conclusion

We learned about the differences between an IAC or EWT service and the SAP GUI for HTML or WebGUI service. Mostly, IAC and EWT services are referred to as Internet services. For such an Internet service templates should be provided in order to define the output of a dynpro to a browser's window. Long ago the parameter `~generatedynpro` was introduced to assist in seldom situations where it was difficult to define custom templates, e. g. for a search help control (F4 help). It makes use of the SAP GUI for HTML framework to render controls and page elements. However, `~generatedynpro` sometimes was utilized *against its intention* to create a service without providing any templates. **The ITS developers recommend to get around `~generatedynpro`!**

In contrast to IAC or EWT services the SAP GUI for HTML provides a full-featured rendering of dynpros with support of menus, function keys, complex controls and also session management. Many users like to benefit from the SAP GUI's user interface and functionality. Now, with introduction of the parameters `~webgui_simple_toolbar` and `~singletransaction` it is easy to create a SAP GUI for HTML based service that meets the requirement of restricting the user from navigation to other programs or transactions.