SAP NetWeaver
How-To Guide

How To... Build Fast and Efficient Dashboards Using the New NetWeaver BW Integration in SAP BusinessObjects Xcelsius

Applicable Releases:

SAP NetWeaver BW 7.01 SP6 and higher, SAP NetWeaver BW 7.02; SAP BusinessObjects Xcelsius 2008, SP2

IT Practice:

Business Information Management

IT Scenario:

Enterprise Reporting, Query and Analysis

Version 1.1

June 2010

HOW-TO GUIDE

THE BEST-RUN BUSINESSES RUN SAP™

Document History

Document Version	Description
1.00	First official release of this guide
1.10	Minor text changes

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Scenario

With the introduction of the new SAP NetWeaver connection in BusinessObjects Xcelsius 2008 we have a direct connection to BW using data from BW queries. Besides query data the connection also offers additional information such as master data values for the characteristics used in the query, actual selected values or variable values. Also filter values and variable values can be set in the connection to filter values at runtime.

In this paper we will give you some directions on how to design your dashboards for optimal performance.

2. Dashboards in General

Xcelsius is designed to be a dashboarding tool. In this respect it is SAP's tool of choice for BI dashboard solutions. It is a key component of SAP's BI strategy and provides a highly visual and interactive user interface.

On the other hand just as every generic tool Xcelsius provides the possibility to build up solutions it is not designed for. Obviously the 'misuse' of the tool can lead to severe performance problems. Thus it is important to understand what dashboards are meant to be and what they are not.

So what is a dashboard? A dashboard is a user interface that provides highly visual and interactive data to the user. It is intended to present data in a simple and intuitive way to the users which are usually executives, managers and operational stuff. In a dashboard it is assumed that the interaction possibilities are defined by the designer of the dashboard. The users follow a predefined path and do not freely navigate through the data. Also the idea of a dashboard is to provide summarized data and key performance indicators to provide an overview of the business performance. So it is rather about giving the big picture instead of going into details.

So what is it not? Xcelsius is not a comprehensive BI tool. It is not a reporting tool for detailed operational reporting (no built-in paging, no export to Excel or PDF), it is not an ad hoc or OLAP analysis tool (analysis paths are predefined by dashboard author) and it is not an application building tool (no state management, no built-in persistence).

Thus as a principle guideline when building up a dashboard you should always keep in mind what a dashboard is intended to be and stick to these ideas.

There is a number of papers and further material on how to build up Xcelsius dashboards in general. In this paper we do not want to repeat the recommendations in detail but want to concentrate on the BW connection. Nevertheless here are just some hints:

Recommendation 1

Restrict the total size of the visual representation according to the UI guideline: "A good dashboard fits on a single screen …"

Do not nest UI containers (Canvas or Panel Containers, Tab Set) more than two levels deep.

Whenever there is a need to use tabs use the label menu button UI element.

Carefully use tab layouts providing dashboards within dashboards, this may lead to slow initialization times because of a high number of visualization components within a single Flash (Xcelsius) file.

3. The BW Integration - General Remarks

The new BW integration offers two main features. On one hand side it provides the possibility to use the BW (and NetWeaver) as the central repository for the dashboards. Thus you can use the NetWeaver transport connection, use the BW translation, and run the dashboards in the BEx Web runtime within the Enterprise portal. You do not need an SAP BusinessObjects Enterprise (BOE) server in this case.

It also offers the possibility to retrieve data from the BW. It is important to note that in the BW connection you can retrieve query data and BW master data (for value help etc) in the same connection. This reduces the number of necessary connections.

The communication between Xcelsius and BW is handled by a new web service. At design time the web service provides query information and stores the dashboard on the BW server. At runtime, the BEx Web runtime provides a link to the SWF file representing the dashboard. The SWF file is loaded by the browser and it calls the web service that provides the BW query data (and master data) in an XML format. The web service accesses the query data via BICS (BI Consumer Services).

Let us now have a look at the steps which are executed once you start a dashboard. In general we can identify four steps that influence the overall response times.

BW Preparation	Flash Loading/Initialization	Web Service Data Retrieval	Flash Processing
Number of connections/BW queries	Complexity of the UI, Excel(*) bindings to connections/UI, Excel(*) logic	BW query runtime, BW web service runtime Result set size	Excel(*) bindings to connections/ UI, Excel(*) logic

Time

(*) **Note**: Excel is only used when designing the dashboard. The logic, bindings and data defined in the spreadsheet are converted into Flex when publishing the dashboard. When viewing the dashboard, the Flex code is executed by the Flash player within the web browser.

In the following sections we will go into more details on how each phase influences the overall dashboard performance. Please note that these steps are ‘idealized’ steps. Some of the steps can be eliminated (BW preparation) or executed at different points in time depending on the design of the dashboard. Some of the steps can also overlap (parallel processing, see below). Nevertheless, we can roughly summarize them as follows:

Total Time= BW preparation + Flash loading/initialization + Web Service + Flash processing

When designing the dashboard one should keep in mind that the business logic and calculation should be designed in the BW layer. The dashboard should just be a presentation of this logic. The rationale behind this recommendation is that the performance of a database engine does not compare with a Flash client application.

As the output of the BW query should on the other hand also be optimized for the dashboard you should not reuse existing queries but rather design query exclusively for the dashboard. Thus we can give a number of general recommendations regarding the used queries:

Recommendation 2

Use queries that are specifically built for your dashboard. Dashboards are not reports so only use a high level pre-aggregated summary which is very different from a report query which needs to show details.

Recommendation 3

Do all calculation and summarization in the query and not in the dashboard.

4. Standard and Stateless Execution of a Dashboard

In order to understand the different modeling options and to see their advantages and disadvantages we have to go a little into the details of how the BW connections use queries. We can distinguish two different ways of executing a dashboard. From the modeling perspective they differ in respect to whether all BW variables are filled from the dashboard itself (and no variable popup should appear) or whether a variable popup should be displayed if a variable is not filled. As a default the system will use the mode where variable popup can be sent. Let us start with this execution mode:

4.1 Execution with Variable Popup Enabled

When a dashboard is started the system determines, as a first action, which queries are contained in the BW connections and instantiates all of them (and creates a data provider for each query). This is done in the step 'BW Preparation'. If the queries contain variables the BW determines whether all variables can be filled automatically, from given URL parameters of the dashboard URL, or whether a variable popup is necessary. Once the variables are filled (if necessary by user entry in the variable popup) these values are stored inside the web service and all data providers are discarded. Each query is then started in a stateless way as soon as the query data is requested (step 'Web service' or later). The variable values will be filled from the storage of the web service. Thus queries can be executed in parallel and also less memory on the java server is necessary.

Please note that all queries contained in the dashboard connections need to be instantiated during the BW preparation time – no matter whether the query data should be read when starting the dashboard or whether the data is only necessary at a later stage. Thus the total number of connections is crucial for the performance of this step.

Important

Before SAPNetWeaver 7.0 EHP1 SP6 Patch 20 the data providers were not discarded but kept throughout the entire execution of the dashboard. Thus we did NOT have a stateless execution of the queries leading to a larger consumption of memory on the Java server. Also a parallel execution of the queries was NOT possible which could cause severe performance issues. Therefore we strongly recommend you to upgrade your system to this patch level and we will concentrate in our further discussion on the behavior shipped with the mentioned patch.

When you are using variables with the same name in different queries then by default in BW these variables will be regarded as one. This concept is known as 'variable melting' or 'variable merging'. Once you change a variable for a query then every query sharing variables with this query needs to be

restarted. This behavior can lead to performance issues and has therefore been changed with the new execution mode introduced with patch 20. As consequence the web service releases the variable melting after the variable popup and avoids unnecessary refreshes which could slow down the system dramatically.

4.2 Execution without Variable Popup

As a new option you can run your dashboards without the variable popup. This modeling option has two implications:

1. No variable popup is sent and no url parameters can be used to set variable values. All variable values must either be filled by default values in the BW or set from the dashboard via each connection. If a variable in a query cannot be filled the system will send an error message once the connection containing this query is executed. Thus the designer of the dashboard has to make sure that necessary variables are always filled.
2. It is not necessary to instantiate the queries once before starting the dashboard. Thus the BW preparation time is reduced to a minimum. Each query is called in a stateless mode only if the corresponding connection is refreshed. Thus the number of connection is not influencing the start performance of the dashboard.

In the following paper we will refer to this execution mode simply as ‘stateless mode’. The stateless mode can be switched on by the URL parameter ‘&XC_MODE=X’.

Note: a stateless execution of an object is an execution where no context is kept by the object. In this paper we use the term in different contexts:

- Query: a stateless execution of a query is an execution where no query context is kept by the called BICS layer.
- Dashboard: we use the word ‘stateless execution’ also for the dashboard as in this case no variable context is kept by the dashboard.

We will now go into the details with the single steps of the execution. Most of the recommendations we give affect not just one but several steps. Usually we will give the recommendation in the first chapter we mention a performance improvement.

5. BW Preparation Time

At the beginning of the execution of the dashboard the standard BW template is loaded. The template is needed as the Xcelsius SWF file runs within the BW web runtime. You do not have an influence on the load time of this step.

If you are using the stateless mode the rest of the preparation time will be reduced to a minimum (no data initialization of the queries) and thus this chapter mainly deals with the case that you are using the standard execution mode.

As we just have explained in the BW preparation the queries contained in all connections in the dashboard will be instantiated. The query definition is read and the system checks whether a variable popup is necessary. Thus even if query data is read at a later stage each query contained in a connection influences the performance.

Recommendation 4

In the standard mode make sure you limit the number of connections: try to use as few connections as possible in your dashboard. Retrieve query data AND master data within

one connection wherever possible. As a rule of thumb 8-10 connections should be the maximum.

In order to reduce the number of connections you can consider using one large query instead of several small queries. Unfortunately using one large query can also lead to a negative effect – if in one step only a small part of the data is to be used the reading of the data for the entire (large) query can lead to a negative effect outweighing the positive effect on the number of connections. Also when using several queries they can be executed in parallel and thus are faster when reading data. You will have to evaluate which design is best for your concrete dashboard. In the stateless mode usually several smaller queries are to be recommended as here the number of connections does not play a crucial role.

The performance of the BW preparation time also depends on the number of used queries or query views. In general you have different options to retrieve data in a connection. Let us assume that you have one query in a connection and you want to retrieve in a second connection some data that is the same set of data but in a different structure (say exchanged columns, characteristics in rows instead of free characteristics etc.). You have three options to do that:

1. Use the same query and change the result transmitted to Xcelsius on the second tab (called 'Data Preview') of the connection in the data manager (let's call it an 'Xcelsius view').
2. Copy the query to a new name and change the query accordingly.
3. Create a BW query view.

These three options are not equal in performance. In general the best performance is reached if 'Xcelsius views' are used. The second best performance is reached if a physical copy of the query is used. The BW query view option is the worst regarding performance.

Recommendation 5

Use as few queries in your connections as possible. Do not use BW query views.

As the query preparation time also heavily depend on the complexity of the query you should make sure that your queries do not contain any overhead and are build as simple and to the point as possible.

Recommendation 6

Keep your queries simple.

6. Flash Loading and Initialization Time

Before the dashboard can be executed the SWF file has to be loaded from the server and initialized. Obviously the SWF loading time depends heavily on the size of the SWF file and thus on the number of objects contained in it. The initialization time also depends on the size and the complexity of the SWF file. In order to understand what is contained in the SWF file we have to go a little into detail with the general design of Xcelsius.

At design time Xcelsius uses Microsoft Excel to model the data exchange between the connections and the screen element. In addition you can use Excel formulas to add additional logic. When you publish the dashboard to BW, Xcelsius generates converts the whole Excel logic into Flex and stores it into the Flash file (.swf). Thus anything contained in the Excel spreadsheet will add up to the size of the SWF file. In addition at runtime all Excel formulas have to be calculated. Thus numerous and complex Excel formulas do not only influence the SWF load time but also the SWF processing time.

Recommendation 7

Keep the number of Xcelsius visualization components reasonable. Search for and remove hidden and unused screen elements.

Recommendation 8

Reduce the number of cells and worksheets used in the Excel spreadsheet.

Recommendation 9

Use Excel formulas as seldom as possible. As recommended above rather do the calculation in the query than in the dashboard. Be especially careful with the use of the 'vlookup' and 'hlookup' functions in Excel as these functions are very time consuming when executed (in the SWF processing). Use the selection "Filtered Rows" of the selector components instead.

When building dashboards it is often very helpful to use dummy data in the Excel spreadsheet as in the BW connection no preview is available. Unfortunately dummy data enlarges the size of the SWF file.

Recommendation 10

Delete or reduce to a minimum dummy data from the Excel spreadsheet before you finalize and publish your dashboard.

With SAPNetWeaver 7.0, EHP1, SP6, Patch 40 the system automatically caches the SWF file on the PC. Thus the loading time (but not the initialization time!) is greatly reduced and the optimizations just mentioned will have a reduced impact on the loading time. Nevertheless the recommendations hold true for the initialization and the processing of the SWF file (see below).

7. Web Service Data Retrieval

In this step the actual BW query is executed and the data is read. Obviously the speed of the query influences this step thus there is one general recommendation:

Recommendation 11

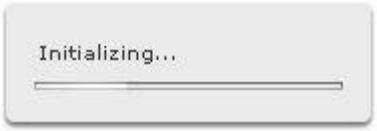
Optimize the read performance of your query by doing the well-known BW performance optimizations such as using aggregates, caching etc.

Obviously a query that is not executed at all does not influence the performance. Thus you should never execute queries on load if you do not know yet whether you will need the data.

Recommendation 12

Only refresh a data connection (and thus reads BW data) when the data is actually needed, i.e. displayed.

Even if the data for a query is needed once the dashboard is started you should consider not using the option 'Refresh Before Components Are Loaded'. If this option is chosen the data will be retrieved before anything is visible in the dashboard. Thus the user can only see the dashboard once the flagged connections are refreshed and will see the message "Initializing…" for quite some time.



Quite often users find it more convenient when they see the (empty) dashboard at once at it gets filled with data as soon as the data is available. You can reach this behavior by assigning a connection to a Connection Refresh Button (potentially hidden) and using the option 'Refresh After Components Are Loaded'. If you use dummy data in your dashboard at design time be sure you delete this data once you are finished designing the dashboard. Otherwise the user will see the dummy data before the real data is refreshed.

Recommendation 13

Replace the option 'Refresh Before Components Are Loaded' with a hidden Connection Refresh button and the option "Refresh After Components Are Loaded"

In the BW connections you also have the option to 'Use Default Data'. You should be very careful when and when not to use this flag. In the paper 'How to efficiently use the SAP NetWeaver BW Integration in SAP BusinessObjects Xcelsius'. In section 4.4 we go into detail when data can be refreshed TWICE if this flag is not properly used.

If you use the option 'Use Default Data' then make sure that a reasonable default value is set. Make sure that no unnecessary data is selected.

Recommendation 14

Be vigilant to correctly use the flag 'Use Default Data'. Be sure that you use proper default values.

Let us again quickly recap how the binding between the result of a BW connection and a screen element in Xcelsius is done. At design time the output of the query in the connection is bound to an Excel range (range 1). Then you specify from which Excel range the display table should pick the data (range 2).

When you execute the dashboard and the connection is refreshed all query data will be read from BW. [As an optimization no query will be executed if no bound range (range 1) is defined.] Now the Web service will return as much data in the XML as fits into the bound range (range 1). [This is an optimization done in Xcelsius 3.1. In earlier versions ALL data will be transferred via XML.] Then the dashboard will fill the data contained in range 2 into the visible table. Obviously the result of the query should not return more data than is transferred to the dashboard runtime and equally range 2 should not be smaller than range 1. [At runtime we will not have any Excel ranges anymore but a flash equivalent but to understand what is happening to the data it is easier to explain it in the model of Excel ranges.]

If the dashboard does not use all data that is returned by the query time is wasted on the query execution, the XML generation and compression, network transfer, XML unpacking, and finally XML parsing.

Recommendation 15

Design your query in such a way that it only returns the data actually needed in the dashboard.

Recommendation 16

Design the range which carries the result of the connection large enough to retrieve all data in the query result.

Recommendation 17

The range for the connection result and the range bound to the screen element should be equal.

In a BW connection you cannot only get query result data but you can also retrieve master data information. The web service will only read master data for a given characteristic if there is an Excel range bound to it. Thus you will have no overhead produced by characteristics that you do not use in the connection. If master data for a characteristic is requested then from Xcelsius 2008, Service Pack 3, Fix Pack 2 on the web service will adapt the number of returned master data entries to the size of the bound Excel range. Thus by setting the size of the range it is possible to control the amount of master data entries returned by the web service. If you are on an older version of Xcelsius the number of returned entries is set to a fixed maximum of 500.

Nevertheless you should make sure that only a reasonable amount of master data is transferred. In a BW query the fixed filter of the query determines how many characteristic values will be available for the value help and thus the number of entries the connection will return. Let us make an example: you use a query in a dashboard to show data for different years. The user can jump from one year to the next by using a drop down box. In order to fill this drop down box you need the value help for the year from the BW connection. If you do not restrict the (global) filter for the characteristic year in the query ALL years maintained in the system will be returned. As usually the dashboard will make sense only for recent years you can restrict the year in the filter (say to 2006-2015) and only 10 values will be returned.

Recommendation 18

Restrict the filter of the query as much as possible.

When a connection is refreshed the query data as well as the master data for the value help is read from the BW and transmitted via the XML. As long as the fixed filter of the query has not changed the returned entries for the value help will be the same as before and a repeated transmission of the master data is not necessary. [The fixed filter of the query can change if you use a variable in the filter and change the variable.]

Recommendation 19

If you use queries that need to be refreshed frequently and large numbers of values in the value help that do not change consider using an extra connection just for the value help. Keep in mind that additional connections have an influence on performance in the standard mode (not stateless).

As a last remark in this chapter we want to come back to a recommendation we have given before. As the data retrieval of the queries is done in a stateless way the web service does not hold any information about the query. Thus anytime a connection is refreshed the query definition will have to be read and the query has to be built up. Thus again – keep your queries simple!

8. Flash Processing Time

Once the data is returned from the web service via the XML the XML has to be decompressed and rendered. Then the logic defined in Excel has to be processed (such as formulas) and the data has to be presented in the screen components. The duration of this step depends heavily on the complexity of the SWF file. For this step we can repeat some of the recommendations that we have given above:

restrict the number of cells used in Excel, restrict the number of Excel formulas, and use 'expensive' Excel function as rarely as possible (vlookup, hlookup).

9. Refresh Options and Parallel Processing

As already described above the system is now capable of execution several connections in parallel. The number of parallel queries obviously depends on the number of parallel connections the browser supports. Microsoft Internet Explorer 7 or earlier is by default set to process two connections in parallel, version 8 is set to 6 connections. It is possible to change that setting with a registry key change on the client machine, please refer to this Knowledge Base article: http://support.microsoft.com/kb/282402.

Recommendation 20

Change the number of parallel sessions in the Internet Explorer. When using another browser please check how many parallel connections are possible.

Obviously changing the number of connections only makes sense if you have several connections in your dashboard that could be refreshed in parallel. You have to be careful not to serialize the connections by the design of your dashboard.

How is this possible? If you are using the result of one connection as an input for a second connection and use a trigger cell that triggers the refresh of the second connections once the first connection has been refreshed then these two connections have been serialized. On the other hand the result of one connection could provide additional information that can be used to restrict the amount of data read in a second connection. Thus your dashboard has be to carefully modeled in such a way, that only a minimum amount of data Is read but also parallelization is possible.

Recommendation 21

Make sure a parallel execution of your connections is possible whenever it makes sense.

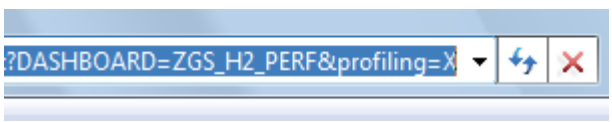
10. Performance Tools

In order to optimize the performance of your dashboard it is crucial to analyze the behavior of your dashboard. There are several tools you can use to analyze the behavior of your dashboard from different points of view.

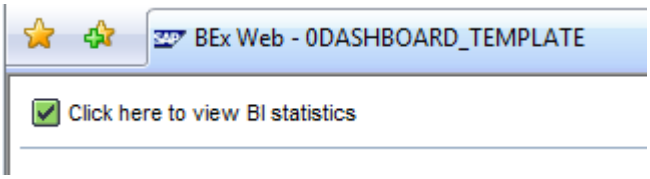
10.1 BW Performance Statistic

The BW offers you some standard options to analyze your performance. As with the new BW connection the BW Web Runtime is used to execute the dashboards these standard performance traces are also available for Xcelsius. It is possible to see what time was spent in the BW Java Stack, what time was spent in the OLAP processor and how much was spent in the data manager.

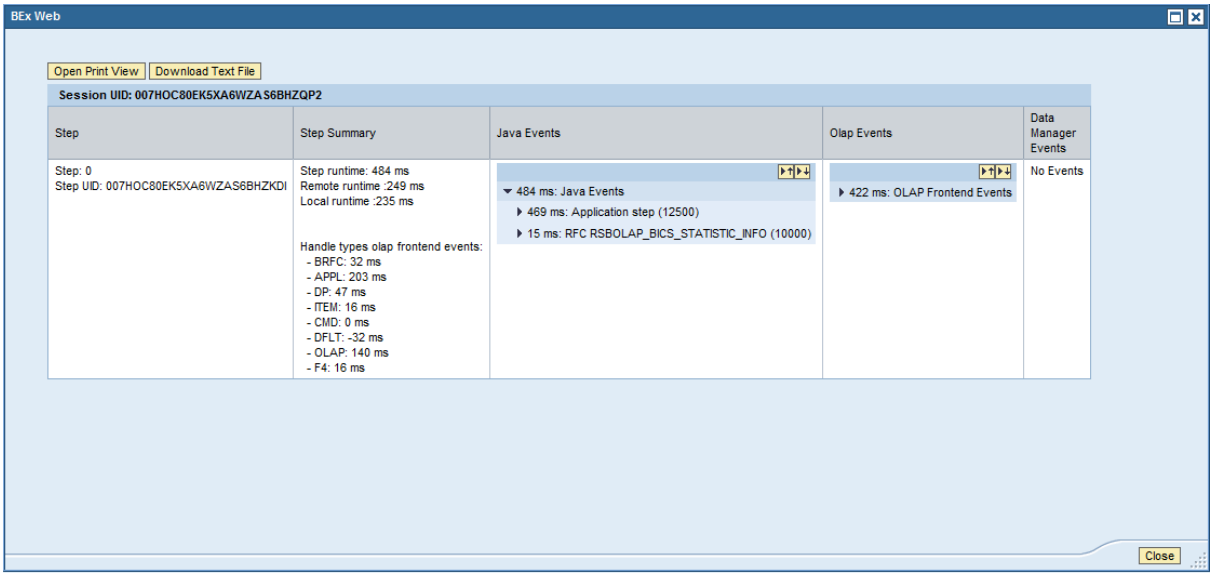
A very easy way to display the BW runtime statistics is to use the URL parameter '&profiling=X'. Just add it to the url of your dashboard and restart.



You will get a link in the dashboard that enables you to start the statistic display.



You will find the familiar BW runtime with the corresponding events, the event-ids and the runtimes.

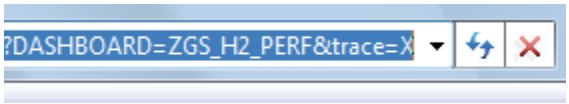


10.2 BW Trace Tool

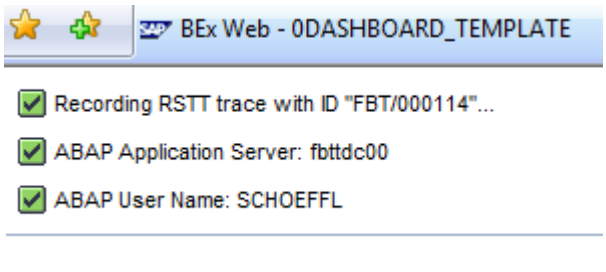
BW also offers a way to trace all calls from the BICS (and thus from Xcelsius) that are performed in ABAP. It enables you to record traces that can be replayed and repeat all user interaction and their results on the ABAP server. From the called function modules you can determine for example how many refreshes were done to the connections.

The trace also gives you some information on the runtime of the function modules and so you can use it to get an idea where your application might have a performance problem.

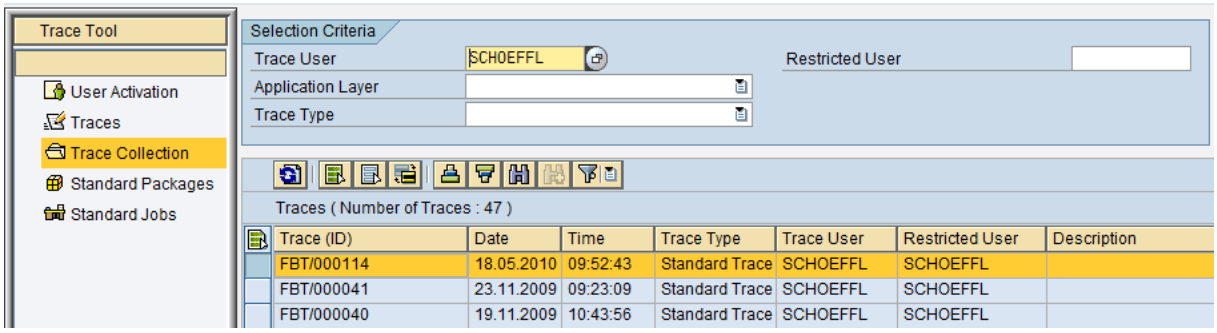
In order to start the recording or view recorded traces use the transaction RSTT in the ABAP backend. You can also start a trace directly from the dashboard. Add the extension '&trace=X' to the url of your dashboard application and restart the application.



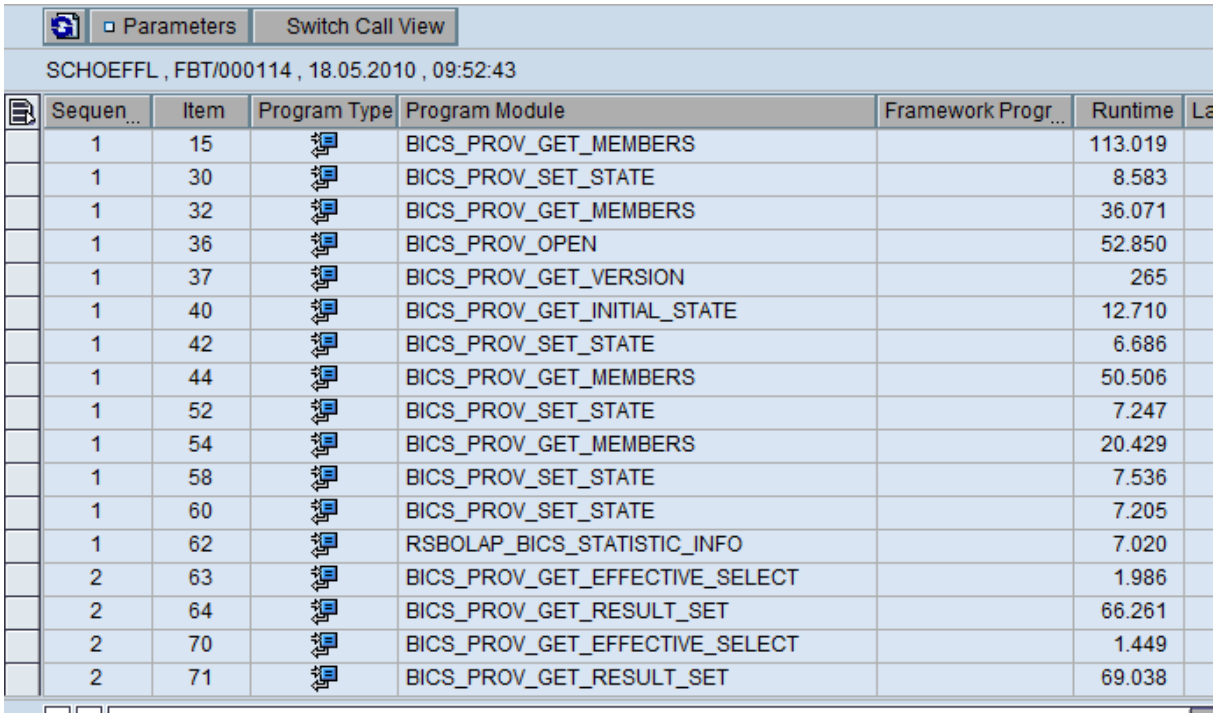
The system will tell you the trace number.



You can now view the trace in the transaction rstt in the ABAP backend system. Make sure you use the right ABAP user which might be different from the Java user used to start the dashboard (depending on the system settings):



You can run the trace for testing or for debugging (e.g. your own coding in the variable exit) or display the trace to get some information about which actions were performed on the backend.



Sequen...	Item	Program Type	Program Module	Framework Progr...	Runtime	La
1	15		BICS_PROV_GET_MEMBERS		113.019	
1	30		BICS_PROV_SET_STATE		8.583	
1	32		BICS_PROV_GET_MEMBERS		36.071	
1	36		BICS_PROV_OPEN		52.850	
1	37		BICS_PROV_GET_VERSION		265	
1	40		BICS_PROV_GET_INITIAL_STATE		12.710	
1	42		BICS_PROV_SET_STATE		6.686	
1	44		BICS_PROV_GET_MEMBERS		50.506	
1	52		BICS_PROV_SET_STATE		7.247	
1	54		BICS_PROV_GET_MEMBERS		20.429	
1	58		BICS_PROV_SET_STATE		7.536	
1	60		BICS_PROV_SET_STATE		7.205	
1	62		RSBOLAP_BICS_STATISTIC_INFO		7.020	
2	63		BICS_PROV_GET_EFFECTIVE_SELECT		1.986	
2	64		BICS_PROV_GET_RESULT_SET		66.261	
2	70		BICS_PROV_GET_EFFECTIVE_SELECT		1.449	
2	71		BICS_PROV_GET_RESULT_SET		69.038	

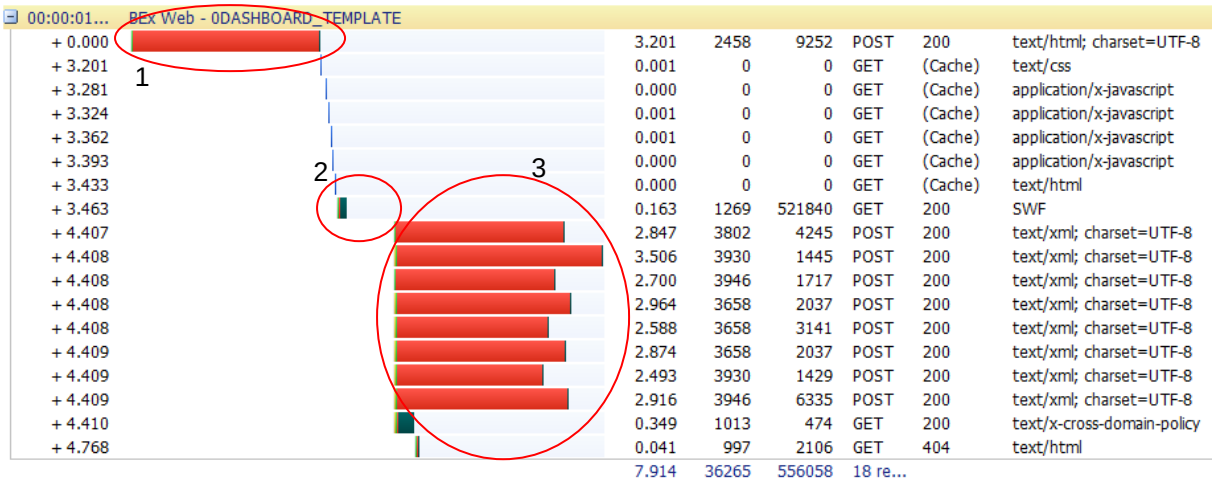
In the above example the last lines show that the result sets (data) for two queries where read (BICS_PROV_GET_RESULT_SET) and also how much time it took. For further information on rstt traces please have a look at the SAP documentation:

http://help.sap.com/saphelp_nw70/helpdata/EN/42/ed1942bcb35433e10000000a155106/frameset.htm

10.3 Frontend Analysis Tools

There are various thirds party tools you can use to analyze the runtime from the browser perspective. We will show you some examples using Http Watch (see www.httpwatch.com). Http watch is an http viewer and debugger that allows you to monitor Http (and Https) without leaving the browser.

Http watch is only tracing the http calls. Rendering time is not shown directly in the trace.



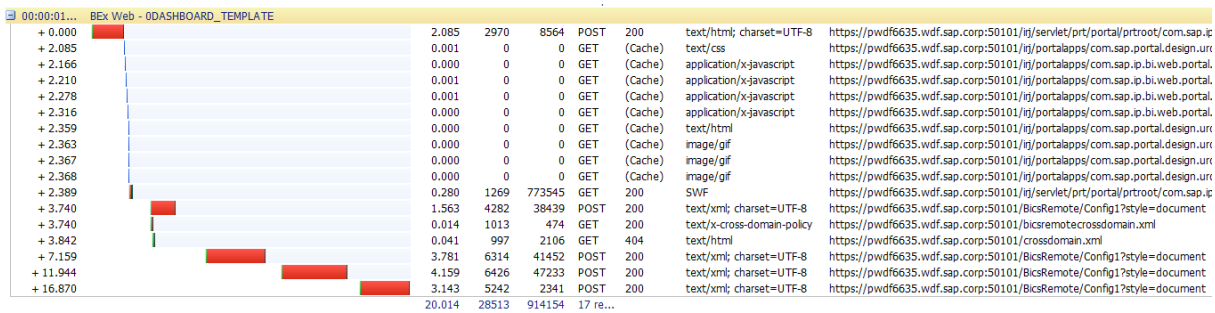
00:00:01...	BEx Web - 0DASHBOARD_TEMPLATE						
+ 0.000	3.201	2458	9252	POST	200	text/html; charset=UTF-8
+ 3.201	0.001	0	0	GET	(Cache)	text/css
+ 3.281	0.000	0	0	GET	(Cache)	application/x-javascript
+ 3.324	0.001	0	0	GET	(Cache)	application/x-javascript
+ 3.362	0.001	0	0	GET	(Cache)	application/x-javascript
+ 3.393	0.000	0	0	GET	(Cache)	application/x-javascript
+ 3.433	0.000	0	0	GET	(Cache)	text/html
+ 3.463	0.163	1269	521840	GET	200	SWF
+ 4.407	2.847	3802	4245	POST	200	text/xml; charset=UTF-8
+ 4.408	3.506	3930	1445	POST	200	text/xml; charset=UTF-8
+ 4.408	2.700	3946	1717	POST	200	text/xml; charset=UTF-8
+ 4.408	2.964	3658	2037	POST	200	text/xml; charset=UTF-8
+ 4.408	2.588	3658	3141	POST	200	text/xml; charset=UTF-8
+ 4.409	2.874	3658	2037	POST	200	text/xml; charset=UTF-8
+ 4.409	2.493	3930	1429	POST	200	text/xml; charset=UTF-8
+ 4.409	2.916	3946	6335	POST	200	text/xml; charset=UTF-8
+ 4.410	0.349	1013	474	GET	200	text/x-cross-domain-policy
+ 4.768	0.041	997	2106	GET	404	text/html
	7.914	36265	556058	18 re...		

In step 1 we see the transmission of the standard BW template from the server to the local PC.

In step 2 we see the SWF loading time and the SWF initialization time. The SWF loading time is the displayed duration of the step. The loading time can be dramatically reduced when using the new SWF caching mechanism (see above). In order to calculate the time needed to initialize the SWF you have to take the start time of the next step (4.407), and subtract the start time of the SWF initialization step (3.463) and the displayed duration of the step (0.163). Thus the rendering time can be seen as the gap visible in step 2.

Step 3 shows the parallel execution of several queries. In this case, the number of connections has been raised and the system can take full advantage of the new parallelization ability.

As a comparison we show the case of a dashboard where by design a parallelization is not possible. In this case each connection is triggered when the result of another connection is available.



00:00:01...	BEx Web - 0DASHBOARD_TEMPLATE						
+ 0.000	2.085	2970	8564	POST	200	text/html; charset=UTF-8	https://pwdf6635.wdf.sap.corp:50101/irj/servlet/prt/portal/prtroot/com.sap.ip
+ 2.085	0.001	0	0	GET	(Cache)	text/css	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.portal.design.ur
+ 2.166	0.000	0	0	GET	(Cache)	application/x-javascript	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.ip.bi.web.portal.
+ 2.210	0.001	0	0	GET	(Cache)	application/x-javascript	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.ip.bi.web.portal.
+ 2.278	0.001	0	0	GET	(Cache)	application/x-javascript	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.ip.bi.web.portal.
+ 2.316	0.000	0	0	GET	(Cache)	application/x-javascript	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.ip.bi.web.portal.
+ 2.359	0.000	0	0	GET	(Cache)	text/html	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.portal.design.ur
+ 2.363	0.000	0	0	GET	(Cache)	image/gif	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.portal.design.ur
+ 2.367	0.000	0	0	GET	(Cache)	image/gif	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.portal.design.ur
+ 2.368	0.000	0	0	GET	(Cache)	image/gif	https://pwdf6635.wdf.sap.corp:50101/irj/portalapps/com.sap.portal.design.ur
+ 2.389	0.280	1269	773545	GET	200	SWF	https://pwdf6635.wdf.sap.corp:50101/irj/servlet/prt/portal/prtroot/com.sap.ip
+ 3.740	1.563	4282	38439	POST	200	text/xml; charset=UTF-8	https://pwdf6635.wdf.sap.corp:50101/BicsRemote/Config1?style=document
+ 3.740	0.014	1013	474	GET	200	text/x-cross-domain-policy	https://pwdf6635.wdf.sap.corp:50101/bicsremotecrossdomain.xml
+ 3.842	0.041	997	2106	GET	404	text/html	https://pwdf6635.wdf.sap.corp:50101/crossdomain.xml
+ 7.159	3.781	6314	41452	POST	200	text/xml; charset=UTF-8	https://pwdf6635.wdf.sap.corp:50101/BicsRemote/Config1?style=document
+ 11.944	4.159	6426	47233	POST	200	text/xml; charset=UTF-8	https://pwdf6635.wdf.sap.corp:50101/BicsRemote/Config1?style=document
+ 16.870	3.143	5242	2341	POST	200	text/xml; charset=UTF-8	https://pwdf6635.wdf.sap.corp:50101/BicsRemote/Config1?style=document
	20.014	28513	914154	17 re...			

Thus the queries are executed in a serialized way.

www.sdn.sap.com/irj/sdn/howtoguides