White Paper

SAP BusinessObjects: Standard Development Kits (SDK)

Abstract  SAP BusinessObjects (BO) is a leading suite of data analysis and reporting tools designed to deliver insight to executives where and when they need it.

Beyond a broad variety of desktop and web applications, BusinessObjects Enterprise provides a Standard Development Kit (SDK) for customized developments.

This paper outlines the possibilities of the SDK and its tremendous value for the BO platform.
Overview
The BusinessObjects Enterprise SDK exposes all functionality known from InfoView and the Central Management Console (CMC) to the developer – a magic wand to those in the know.

The SDK is available for the two prominent development platforms, Java and .NET, using the same programming interface (API) in both cases, thus ensuring the use of the SDK in virtually all environments.

Here is an example: the InfoStore object in both worlds provides access to the BusinessObjects Enterprise (BOXI) objects stored in the Central Management Server (CMS) repository, a relational database.

In the following sections we explain where the use of the SDK is appropriate and which part of the SDK is used best.

Available SDKs
The following SDKs are available:

- Enterprise SDK (Java & .NET)
- Report Application Server (RAS) SDK
- Report Engine SDK (Java & .NET)
- Web Service SDK (Java & .NET)
- Universe Designer SDK (COM)
- Desktop Intelligence SDK

The Enterprise SDK provides the means to manage platform related processes such as authentication, user management, role management, report management, report scheduling and access control. Roughly speaking, it covers all functionality of the CMC web application.

The following Java example shows how to log on to the BusinessObjects platform.

Example 1: Logon using the Enterprise SDK.

```
ISessionMgr sm = CrystalEnterprise.getSessionMgr();
IEReportSession enterpriseSession = sm.logon(“username”, “password”,
“server:port”, “secEnterprise”);
IInfoStore infoStore = (IInfoStore)session.getService(“InfoStore”);
// … working code follows
enterpriseSession.logoff();
```

The Report Application Server (RAS) SDK is used to create and modify Crystal reports at runtime. It is not further discussed in this paper.

The report engine SDK is used to view and modify Desktop Intelligence and Web Intelligence documents and to provide reports in different formats (.xls, .csv, .pdf, .html and so forth). This functionality is often used to provide BO reports to other applications, e.g. the intranet portal.

The report engine SDK does not work with Crystal reports – these reports are covered by the Report Application Server SDK mentioned above.

The Web Service SDK is used for client applications and – being based on http and SOAP – easily connects through complex firewall infrastructures. It provides interfaces to the Enterprise SDK and Report Engine SDK. Again, the Crystal Reports related SDK (RAS) is not supported.

Using a web service to logon is very similar to example 1.

Example 2: Logon using a web service

```
Connection connection;
URL boConURL = new URL(“<BOXI>/dswbobje/services/Session”);
connection = new Connection(boConURL);
Session boSession = new Session(connection);
EnterpriseCredential credential =
EnterpriseCredential.Factory.newInstance();
credential.setLogin(“username”);
credential.setDomain(“domain”);
credential.setPassword(“password”);
SessionInfo boSI = boSession.login(credential);
```

SDK on Duty
D1 Solutions has a long track record in using the SDK to solve demanding problems to the benefit of our customers. Examples are: system analysis, system migration, and bug fixing. The following section discusses possible situations and how we at D1 Solutions use the SDK in our daily work.
How do you achieve to compare different BO XI instances? This may be necessary during a migration to a new sever or just to synchronize the state of different BO XI instances such as production, testing or development. Items to compare could be: roles and permissions, scheduling, event triggers, and logging configurations.

The following example shows how to retrieve a list of reports using the Enterprise SDK: an SQL like query language is used to retrieve object information from the repository database.

```
String query = 
    "SELECT * FROM CI_INFOOBJECTS 
    +" WHERE SI_INSTANCE = 0";
IInfoObjects manyReports = 
    infoStore.query(query);
IInfoObject report = (IInfoObject) 
    manyReports.get(0);
System.out.println("Report: "+ 
    report.getTitle());
```

Example 3: Query a list of all reports.

Imagine you have over hundred daily recurring schedules and you want to edit or replace these schedules. The SDKs may help you first to document the current schedules and afterwards to update or replace schedules where needed. Without the use of the SDK, this would be a time consuming (and, to be honest: boring) task in the CMC. Example 4 shows how to schedule a Web Intelligence report in PDF format.

```
//IInfoObjects webiReports;
IInfoObject report = 
    (IInfoObject)webiReports.get(0);
IWebiFormatOptions reportFormatOptions = 
    ((IWebi)report).getWebiFormatOptions();
reportFormatOptions.setFormat(CeWebiFormat.PDF);
ISchedulingInfo scheduleInfo = 
    report.getSchedulingInfo();
scheduleInfo.setType(CeScheduleType.ONCE);
scheduleInfo.setRightNow(true);
infoStore.schedule(webiReports);
```

Example 4: Setting a schedule.

If you redesign objects in your universe, it is important to know which reports are affected by the change. The SDK may help you to get information about how universe objects are used in reports. For example, you may have a complex filter object in your universe filtering some product groups and countries.

Now, as your customer is successful, you need to add additional countries to the report. Here, the impact analysis shows you which reports are using the country filter and thus tells you if any (and if so, which) reports depend on that filter.

There are many reasons why a list of reports needs to be changed – e.g. because a new logo or new wording must be used. Without the SDK, you have to open each report, do the changes manually and save the report again. Using the SDK, you may develop an application which does the update of the reports for you.

If you delete a user from BO XI, his personal folder and all his reports get deleted as well. A customized process could copy such reports to a separate section before deleting the account. Such processes may improve operational tasks and help to ensure a good integration of the BO XI administrative processes with your customers way of doing things.

Business Objects security concept is very powerful. This also implies that it is demanding, both in set-up and in maintenance. Both tasks need a lot of (again, rather boring) interaction with the Central Management Console (CMC).

Such mass mutations of objects in the BO XI repository are easily done with the help of the SDK.
A Swiss Army Knife for BusinessObjects
To the benefit of your customers, at D1 Solutions we have developed a highly customizable and extensible toolkit for BusinessObjects Enterprise to help us with repetitive tasks, named DBOXS.

User and user groups, reports and folders are managed based on simple CSV lists, and data mutations are written directly to the CMC repository.

This saves time during operational tasks and improves process excellence with the elimination of error prone manual mass mutations.

Discussion
To those in the know, the BO SDKs provide power beyond the regular BusinessObjects applications and help to handle consuming processes efficiently during development and operation of a BOXI platform.

Also, given the strong integration features, they help to bring the report content closer to the decision makers (e.g. with the portal integration). D1 Solutions is well experienced in analyzing demands, defining requirements and developing sophisticated solutions based on the BO SDKs.

Our dedicated toolbox DBOXS supports our consultants in all the discussed scenarios on a daily basis.

Author
Adrian got a degree in earth science from University of Basel in 2001 and added a Master of Business Administration from Swiss Federal Institute of Technology Zurich (ETH) in 2006. His areas of expertise are process optimization, visualization of management information, and client analytics. Projects in financial services, utilities, and retail. Adrian has been in the team since 2008.

Your contact for further information
Simon Hefti, Chairman
simon.hefti@d1-solutions.com, P +41 44 435 10 10