

RBD/RTC INTEGRATION

The ASIST Rational Business Developer and Rational Team Concert integration solution is a tight coupling between development, automated Builds, Release Management and Configuration Management.

Since the announcement of Rational Team Concert (RTC), ASIST started to build an integration with Rational Business Developer (RBD/EGL). The purpose of this integration is :

1. To have a version control system for EGL
2. To have Configuration Management in place for different environments (Development Environment, Accept Environment, Production Environment or any other configuration or combination)
3. To have a tight coupling between Release Management, EGL Builds and Deployment independently of the target system (System z, System i, Web projects, Services, RUI/Web 2.0)

Our integration solution is independent of the required RTC version, so it can be integrated from RTC Express C till RTC Enterprise Edition and is a best practice framework that covers end-to-end setup of RTC Streams till automated build and deployment.

**Development Lifecycle**

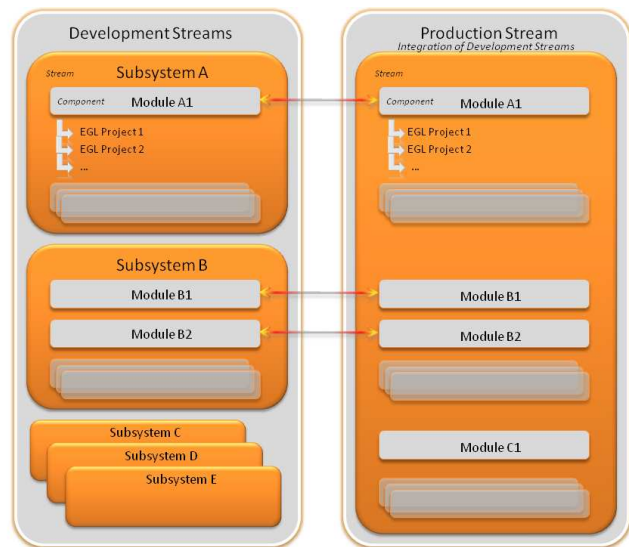
To be able to make changes developers create a personal 'Repository Workspace' of the Stream (subsystem) they want to make changes for.

Once a change is considered completed it can be 'Delivered' by the developer from his repository workspace to the according stream. This is done by delivering the change shown in the 'Pending Changes' view. This view will show all delta's between the 'Repository Workspace' and the flow target which is defaulted to the DEV Stream.

Once delivered, the DEV Stream will then submit this new change to the other developers and give them the possibility to accept the change into their own repository workspace to update it to the latest know state.

After testing we now want that change to be put into production.

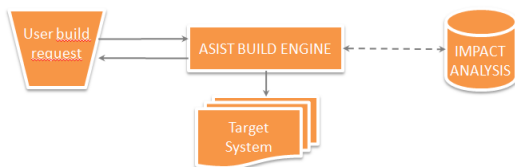
To achieve the delivery of the change to the PRD Stream we change the flow target of our current 'Repository Workspace' to the PRD Stream. Once the flow target got changed, the 'Pending Changes' view will show all deltas between the user's repository workspace and the PRD Stream. The change(s) to be delivered to the PRD Stream can be selected. RTC gives us the opportunity to request approval before delivering. This approval will ensure the consistency of the PRD Stream and gives another level of security. Once in the PRD Stream the change will be available for the production build engine.



**RTC Organization**

Important in such an integration is the organizational setup of the Projects, Components and Streams in RTC. To maximize the performance we are limiting the amount of files loaded into the RBD workspace, we opted to divide the DEV Stream in by subsystem. This results in a number of DEV Streams. Each one of these subsystem Streams is divided in one or more Components. Those Components reflect the subsystems modules and each Component contains the EGL source.

The PRD Stream, which in fact will act like an integration Stream, will not be divided by subsystem. But instead it will be kept as a whole representation of the version of the sources in Production. However, this PRD Stream will contain the same components as the ones existing under the several DEV Streams. This is necessary for delivery from DEV to PROD.



**ASIST EGL Build Engine**

The current Jazz Build Engine does not support generation of EGL artifacts like Programs, Services, JSF, RUI Handlers etc.

The ASIST EGL Build Engine is developed as an extension to the standard Jazz Build Engine and consists of an integrated ANT-script and a Build Server program that automates the builds of the EGL artifacts. To kick off EGL Builds we make use of the standard EGLCMD who's XML input file gets generated dynamically from the Build Server program which discovers the differences (delta's) between Streams to build up that EGLCMD XML input file automatically.

Optionally we integrate our ASIST Impact Analysis tool to drive Smart Build or at least to get informed which impact some changes have on all the EGL artifacts so their Build can be requested by the Build Manager.

## Impact Analysis Tool

In EGL we develop Programs, Libraries, Services, Forms, Databases, JSF and RUI Handlers. All of these EGL artifacts are generatable into executables and are built-up with Records and Functions. These Records and Functions are often commonly used by more than one EGL artifact. EGL is a 4th generation language that generates Cobol or Java, which in their turn, get compiled to an executable module. The fact that some EGL parts are commonly reused by more than one generatable artifact has consequences when we want to change such a part. It impacts the other generatable artifacts as they might malfunction when the change is not managed correctly.

So when changing a common part we should be aware of the fact that we, probably, need to re-build all the artifacts that are using that part in order to keep the executables in synch with the changes we made to that common part. In some cases we don't need to re-build, so an automated build of impacted generatable parts is not always the right solution.

Rational Business Developer, Rational Team Concert, Jazz and EGL are trademarks or registered trademarks of IBM Corporation

## Lexicon

### Source Control Management

Source Control keeps track of the different changes that are made to a source. This can be expressed as being different versions. The Source Control feature provides the user the possibility to manage the various versions of source code

### Configuration Management

Configuration Management keeps track of source code versions together with their associated runtime environments.

### Source and Configuration Management (SCM) tool

An SCM Tool supports both Source Control Management and Configuration Management

### Lifecycle Management (Configuration Management tool).

The tool for Lifecycle Management, also referred to as Configuration Management, supports SCM and has features to trigger and manage the Build process to the different runtime environments

### Application Systems and Subsystems

Application Systems and Subsystems are the logical split of sources that functionally belong together. Sometimes Systems can be defined so big that they be better split into multiple Subsystems to have a better overview

### Release Management

Some companies follow a strict Release Calendar for putting their projects and changes into production. This Release Management driven approach is nicely integrated in RTC

### Repository Workspace

Repository Workspace is an implementation of RTC that offers an asynchronous personal server site workspace, which can be detached from the local Workspace to work offline

## Key Benefits



- Quick Start to the integration of RBD development
- Quick Start to the RTC usage and implementation
- Important time savings
- 100% customizable around specific customers needs



## ASIST - Leading Application Development & Integration

In less than a decade, ASIST became a leading international player in the area of Application Development for large organisations and institutions. Every day ASIST teammembers are deploying their services at respected international organisations.

Our vision is to develop and maintain leadership in integrating people, processes and information technologies. Our customers are coming from all kinds of industries and government institutions, and are asking us to integrate their current legacy with new technologies. Learn more on our customer experiences