

User's Guide for Server

REQUBE REPORT Server 6.0

RQSERVER





REQUBE®

리큐브 리포팅 솔루션으로 Quick & Quality! 두마리 토끼를 잡으십시오!

REQUBE REPORT는 친숙하고 간편한 사용자 인터페이스로 어떠한 데이터 소스와도 연계되어 정교하고 화려한 보고서 양식을 쉽게 작성할 수 있도록 하는 리포팅 솔루션입니다.



► Any user! Easier to use!

MS Office UI에 익숙한 누구든지 쉽게 사용할 수 있습니다.

- ► Wherever DATA is available!
 어떠한 DBMS. 미들웨어, 텍스트파일에도 접근이 가능합니다.
- ► Any Report Possible!
 어떠한 복잡한 폼과 형식의 리포트도 개발이 쉽습니다.
- ► High Performance!

Enterprise급 대용량 보고서 처리가 가능합니다.

► Light & Robust Engine, Flexible Server

가볍고 견고한 엔진, 확장성 높은 서버로 안정성을 보장합니다.

REQUBE REPORT SERVER 를 시작 하기 전에	•••••	9
제품 특징		Ć
설 치		
디렉토리	1	4
문서 실행 과정	. 1	5
1. 서버 관리자	. 1	7
1.1 관리 정보	1	7
1.2 로그인	. 1	8
1.3 문서 관리	. 1	8
1.3.1 폴더 이동	1	8
1.3.2 폴더 등록	1	9
1.3.3 폴더 삭제	1	9
1.3.4 문서 등록	2	C
1.3.5 문서 삭제	2	1
1.3.6 데이터베이스 설정	2	1
1.4 사용자관리	. 2	2
1.5 데이터베이스	. 2	2
1.6 설치관리	. 2	3
1.7 환경변수	. 2	4
1.8 로그모니터	. 2	5
2. 문서 실행	. 2	5
2.1 제품 구성도와 실행 메카니즘	. 2	6
2.2 문서 실행 JSP	. 2	7
2.3 RQVIEWER.JSP의 구성	. 2	7
3. RQVIEWER	. 3	C
3.1 구조	. 3	С
3.2 API 함수	. 3	1
3.2.1 뷰어(RQViewerCtrl)	3	1
전체 함수 사용 예제	3	1
HTTP 호출 JSP 예제 (rqviewer_http.jsp)	3	1
RQViewerCtrl - HTML <object> 코드 (./setup/rqviewer.js)</object>	3	6
하스 기느 저기	2	c



OpenReport	3 6
CreateConnectionControl	3 7
Run	3 7
CreateDataSetControl	3 8
GetReport	3 8
GetConnectionControl 함수	3 8
GetDataSetControl 함수	3 9
CloseReport	3 9
Stop	3 9
Refresh	4 0
MoveFirtstPage	4 0
MoveLastPage	4 1
MoveNextPage	4 1
MovePrevPage	4 2
MovePage	4 2
ZoomOut	4 3
ZoomIn	4 4
Zoom	4 4
ShowToolBarButton	4 5
Print	4 7
PrintDialog	4 8
PrintPreview	4 9
SaveAsDialog	4 9
SaveAs	5 0
SendToExcel	5 0
SendToPDF	5 1
SetBaseURL	5 2
CrossTabPageBreak	5 2
GetDocType	5 3
GetTotalPageNo	5 3
GetCurrentPageNo	5 4
속성	5 4
BackColor	5 4
ToolBarVisible	5 5
StatusBarVisible	5 5
ShowProgressDialog	5 6
Empty Data Check Ontion	5.6



	이벤트 정리	. 5	
	EndRunReport 이벤트	. 5	7
	EmptyResultData 이벤트	. 5	7
	Refresh 이벤트	. 5	8
	ButtonSaveAsClickBefore 이벤트	. 5	Ć
	ButtonSaveAsClickAfter 이벤트	. 6	(
	ButtonPrintClickBefore 이벤트	. 6	(
	ButtonPrintClickAfter 이벤트	. 6	1
	ButtonRefreshClickBefore 이벤트	. 6	
	ButtonRefreshClickAfter 이벤트	. 6	2
	ButtonStopClickBefore 이벤트	. 6	(
	ButtonStopClickAfter 이벤트	. 6	2
	ButtonZoomOutClickBefore 이벤트	. 6	2
	ButtonZoomOutClickAfter 이벤트	. 6	ļ
	ButtonZoomInClickBefore 이벤트	. 6	6
	ButtonZoomInClickAfter 이벤트	. 6	6
	ButtonMovePrevPageClickBefore 이벤트	. 6	-
	ButtonMovePrevPageClickAfter 이벤트	. 6	8
	ButtonMoveNextPageClickBefore 이벤트	. 6	8
	ButtonMoveNextPageClickAfter 이벤트	. 6	Ç
	ButtonMoveFirstPageClickBefore 이번트	. 7	(
	ButtonMoveFirstPageClickAfter 이벤트	. 7	
	ButtonMoveLastPageClickBefore 이번트	. 7	
	ButtonMoveLastPageClickAfter 이벤트	. 7	2
3	.2.2. 리포트 객체(RQReport)	7	(
	함수 기능 정리	. 7	(
	SetRuntimeVariable	. 7	(
	GetRuntimeVariable	. 7	(
	GetObject	. 7	(
	GetSQLControl	. 7	2
3	.2.3 리포트 컴포넌트 객체(RQObject)	7	4
	함수 기능 정리	. 7	2
	SetProperty	. 7	2
	GetProperty	. 7	ļ
3	.2.4 SQL 컨트롤 객체(RQSQLControl)	7	Ę
	함수 기능 정리	. 7	ļ
	CatCount	7	Ĺ



GetDefaultRowSeperator / GetDefaultColSeperator	
GetDBString	7 6
GetSQLString	7 6
GetQueryCount	7 6
GetDBIFType	7 7
GetXMLRoot	7 7
GetConnectionString	7 7
GetDBInfo	7 8
GetSQLStmt	7 8
GetSQLStmtCount	7 8
GetDBID	7 9
GetRowSeperator	7 9
GetColSeperator	7 9
GetEncoding	8 0
3.2.5 RQConnectionCtrl 객체	8 0
다중 Interface 전체 예제(rqviewer_multi.jsp)	8 0
함수 기능 정리	8 6
AddConnection	8 6
GetConnection	8 7
GetConnectionByName	8 7
GetConnectionByID	8 8
GetConnectionType	8 8
Send	8 9
GetCount	8 9
3.2.6 RQConnectionHTTP 객체	8 9
HTTP REQUBE Server 연결 예제 (rqviewer_http.jsp)	9 0
HTTP XML 연결 예제(rqviewer_http_xml.jsp)	9 4
함수 기능 정리	9 9
GetPath	9 9
SetPath	9 9
AddParameter	1 0 0
GetParameter	1 0 0
GetResponseFile	1 0 1
GetEncoding	1 0 1
RemoveParameter	1 0 1
GetCount	1 0 2
ReplaceParameter	1 0 2



SetQueryCount	1 0 2
SetEncoding	1 0 3
GetURL	1 0 3
AddResponseCount	1 0 3
GetResponseIndex	1 0 4
GetResponseType	1 0 4
SetResponseType	1 0 4
SetXMLRoot	1 0 5
GetXMLRoot	1 0 5
SetSeperator	1 0 5
GetRowSeperator	1 0 6
GetColSeperator	1 0 6
3.2.7 RQConnectionJDBC/RQConnectionSAP 객체	1 0 6
함수 기능 정리	1 0 6
Response	1 0 6
GetEncoding	1 0 7
GetResponseFile	1 0 7
GetCount	1 0 7
SetDBString	1 0 8
SetQueryString	1 0 8
SetQueryCount	1 0 8
SetDBInfo	1 0 9
SetSQLStmt	1 0 9
GetResponseIndex	1 0 9
GetResponseType	1 1 0
SetResponseType	1 1 0
AddDBInfo	1 1 0
AddSQLStmt	1 1 1
AddResponseCount	1 1 1
3.2.8 RQConnectionLOCAL 객체	1 1 1
LOCAL 연결 예제 (rqviewer_local.jsp)	1 1 1
함수 기능 정리	1 1 7
GetEncoding	1 1 7
GetResponseFile	1 1 7
GetCount	1 1 8
AddResponseFile	1 1 8
SetEncoding	1 1 8

	AddXMLResponseFile	. 1	1	9
	AddXMLResponseString	. 1	1	9
	GetXMLRoot	. 1	1	9
	GetResponseIndex	. 1	2	0
	GetResponseType	. 1	2	0
	SetResponseType	. 1	2	0
	AddResponseString	. 1	2	1
	SetSeperator	. 1	2	1
	GetRowSeperator	. 1	2	1
	GetColSeperator	. 1	2	2
3	3.2.9 RQDataSetCtrl 객체	1	2	2
	함수 기능 정리	. 1	2	2
	AddDataSet	. 1	2	2
	GetDataSet	. 1	2	2
	GetCount	. 1	2	3
	SetDataSet	. 1	2	3
3	3.2.10 RQDataSetTXT 객체	1	2	3
	함수 기능 정리	. 1	2	4
	SetDataSetFile	. 1	2	4
	SetEncoding	. 1	2	4
	SetSeperator	. 1	2	4
	GetRowCount/GetColCount	. 1	2	5
	GetValue	. 1	2	5
	SetDataSetType	. 1	2	5
3	3.2.11 RQDataSetXML 객체	1	2	6
	함수 기능 정리	. 1	2	6
	SetDataSetFile	. 1	2	6
	SetBindingOption	. 1	2	6
3	ADI 하스 이데人	1 '	2	7



기업의 보고서 저장소인 REQUBE REPORT Server 는 사용자가 웹 브라우저에서 데이터베이스 정보를 실시간으로 분석하고 보고서를 출력할 수 있도록 한다. 수많은 사용자가 웹에서 동시에 문서를 출력하는 실행 기능과 사용자정보 공유 및 다양한 API 를 통한 타 시스템과의 통합 등 강력한 서버 기능으로써 효율적인 정보 인프라를 구축할수 있다.

□ 웹 환경의 어플리케이션 개발 시 Free Style 형태의 보고서 실행 및 출력이 가능

행과 열로 구분되는 표 형태의 정형적인 데이터베이스 정보 검색뿐만 아니라 워드 프로세서로 작성된 문서와 같은 자유 형식의 보고서 출력이 현장에서 요구되고 있다. REQUBE REPORT Server 는 REQUBE REPORT Designer 에서 설계한 문서를 REQUBE REPORT Server에 등록하고 실행함으로써 일반 사용자들이 웹 브라우저만 가지고도 윈도우에서 실행한 것과 같은 복잡하고 비정형적인 보고서 내용을 얻을 수 있으며, 인쇄나 저장이 가능하다.

■ HTML 문서에 REQUBE REPORT 문서를 간단히 통합

REQUBE REPORT Server 는 REQUBE REPORT Designer 에서 작성된 문서를 HTML 문서에 삽입하는 기능을 제공한다. 이는 결과 문서가 아닌 REQUBE REPORT Designer 의 설계 문서를 HTML 문서에 삽입한 것으로, HTML 문서가 실행될 때 REQUBE REPORT 문서가 함께 실행되어 그 결과를 동적으로 보여진다. REQUBE REPORT 문서 실행을 요청시 HTML 폼이나 JSP 등에서 다양한 실행 조건을 부여함으로써 동적인 문서 통합이 가능하다.

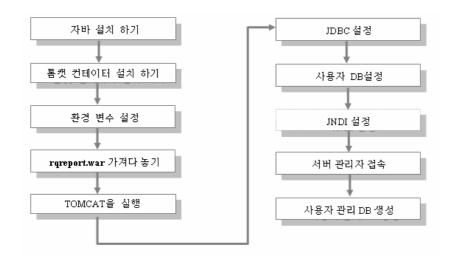
▶ 사용자별/그룹별 REQUBE REPORT 문서 관리

개인별 혹은 업무 그룹별로 사용자와 그룹을 등록하여 서버에 등록한 REQUBE REPORT 문서에 대한 정보 접근 권한을 부여하고 통제함으로써 데이터베이스 정보 보안 및 공유를 원활이 하여 업무 생산성이 향상된다. 또한 대규모 사용자 및 문서를 효율적으로 관리할 수 있는 그룹/폴더 기능도 제공된다. 이러한 기능은 REQUBE REPORT Server 설치 시 함께 제공되는 서버 관리자 프로그램을 이용하여 손쉽게 구현할 수 있다.

▶ API 를 통한 통합 시스템 구축

사용자의 각 시스템 요구에 맞게 효과적으로 사용자 관리, 문서관리, 데이터베이스 정보관리를 할 수 있도록 다양한 서버 API가 제공된다. 또한 REQUBE REPORT Server 내의 사용자, 그룹과 타 시스템의 사용자, 그룹 데이터베이스를 연계하는 API를 제공함으로써 시스템간의 통합을 용이하게 할 수 있다.

REQUBE REPORT Server 를 사용하기 위해서는 다음과 같이 설치과정이 순차적으로 이루어져야 한다. REQUBE REPORT Server 은 WAR 형태로 제공되며 WAS(Web Application Server)에 설치되어 동작한다. 데이터베이스 연결은 WAS 에 등록된 JNDI을 사용한다.



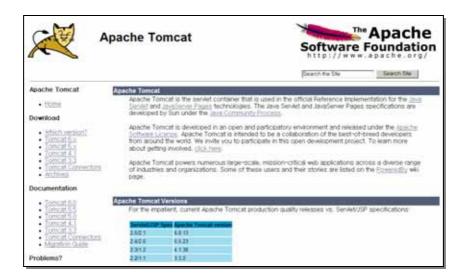
톰켓 설치과정은 다음과 같다.(이 설치 과정은 Tomcat을 기준한 것이며 Tomcat 버전은 5.0.28 이다.)



http://java.sun.com

① 먼저 System에 JDK(Java Development Kit)가 없는 경우 JDK를 설치한다. 버전은 1.4.x 이상의 설치를 권장한다. (버전에 상관없이 설치해도 무관하지만 1.4 이상 설치를 권장한다.)





http://tomcat.apache.org

🥝 System 에 WAS 가 없는 경우 WAS를 설치한다. Tomcat 버전은 5.0.x 설치를 권장한다.

(Tomcat 버전은 관계없으며 단지 권장 사항일 뿐이다)

◎ 환경 변수 설정을 한다.

[내컴퓨터]-[속성]-[고급]-[환경변수] 중 JAVA_HOME, TOMCAT_HOME, classpath 시스템 변수에 값을 지정한다.

 $JAVA_HOME$: Java 가 설치된 폴더 지정

예> C:\j2sdk1.4.2_12



CLASSPATH

예> .;%JAVA_HOME%\jre\lib\rt.jar



TOMCAT_HOME: TOMCAT 이 설치된 폴더 지정

예> C:\Program Files\Apache Software Foundation\Tomcat 5.0



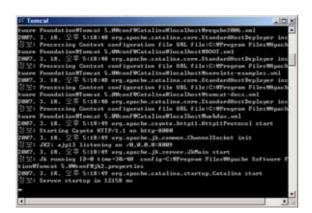




🔍 %TOMCAT_HOME% 는 TOMCAT이 설치된 폴더를 %JAVA_5.0%는 JAVA가 설치된 폴더를 의미한다.

REQUBE REPORT Server 제품 설치과정은 다음과 같다.

💶 디플로이(deploy)를 위해 rqreport.war 파일을 %TOMCAT_HOME%\webapps 폴더에 복사한다.



TOMCAT 을 실행한다.

실행파일 위치는 %TOMCAT_HOME%\bin\startup.bat (Unix 에서는 startup.sh)이다. 이 파일을 더블 TOMCAT 이 실행되고 rqreport.war 파일이 디플로이 (autodeploy)되면서 webapps 폴더 아래에 rqreport 폴더가 생성된다.

🔍 autodeploy : 각 WAS 별로 디플로이 방법에 차이가 있으므로 각 WAS 환경에 맞게 디플로이를 한다. 기본적으로 TOMCAT 은 autoDeploy="true"로 설정이 되어 있어 webapps 에 rereport.war 파일을 가져다 놓으면 자동 디플로이가 된다

관리 DB 설정은 다음과 같다.

JDBC 드라이브를 설정한다.

[ojdbc14.jar 파일 복사]

[CLASSPATH 설정]

(각 WAS 설정파일에서 설정해 준다.)

원본 파일 경로 :[잘나내기]

%TOMCAT_HOME%\ webapps\ rqreport\WEB-INF\lib\

복사 파일 경로 :[이동]

%TOMCAT_HOME%\ common\lib

※ TOMCAT 일경우 ojdbc14.jar 파일을

%TOMCAT_HOME%\ common\lib 로 복사하지말고 이동

추가위치: "set CLASSPATH=" 뒷부분에 추가

🥝 사용자 관리 DB 설정하기 🗕 REQUBE REPORT Server 의 관리 정보를 저장할 사용자 DB 를 설정한다.

[rqreport.properties 에 값 설정]

사용자 관리 DB 의 jdbc-Driver 와 connection String 지정:

rqreport.repository.jdbcDriver=sun.jdbc.odbc.JdbcOdbcDriver rqreport.repository.connStr=jdbc:odbc:Repository

Repository 폴더 지정(문서를 등록했을 경우 실제 저장되는 경로로 사용자 직접생성가능):

rgreport.repository.path=C:\test

(각 OS 에 맞게 지정된 위치에 폴더를 생성해 주는 것이 좋다.)

임시 로그인을 위한 id/pw 확인 :

rqreport.repository.install_id=install

rgreport.repository.install_pw=test

rqreport 최초 설치시 관리 DB 가 아직 생성되지 않은 상태이기 때문에 관리자 로그인이 불가하므로 인스톨을 위한 임시 id/pw 로 로그인한 뒤 관리 DB 생성을 하고 REQUBE PEPORT Server 테이블 생성이 끝난 뒤에는 위의 초기 설치시 필요한 id/pw 이였으로 id/pw 설정을 삭제해 주는 것이 좋다.

③ 만약, 설정된 사용자 관리 DB 가 JNDI 에 설정되어 있지 않은 경우에는 JNDI 설정을 위해 아래와 같이 입력 후 [Login]버튼을 클릭한다.

http://localhost:8080/admin/

(server.xml, catalina 폴더 안에 있는 rqreport .xml 에 직접 수정해 주어도 되나 WAS 별로 제공하고 있는 어드민 툴을 이용하여 JNDI 값을 지정한다.)

- ◎ 톰캣 어드민 툴에서 [Resource]-[Data Source]]을 클릭 후 사용자 관리 DB 의 다음 항목을 지정하고 [확인]버튼 을 클릭한다.
 - JNDI Name
 - Data Source URL
 - JDBC Driver Class
 - User Name
 - Password



JNDI Name:	[jdbc/RGreqube	
Data Source URL	jdbc:odbc:WorldAV -	
JDBC Driver Class	sun.jdbc.odbc.JdbcOdbcDriver	
User Name.		
Password		
Max. Active Connections:	[4	
Max. Idle Connections:	2	
Max. Wait for Connection:	5000	
	3	
Validation Query.	-	



- S REQUBE REPORT Server 관리자를 실행한다. Explorer 의 주소창에 아래와 같이 입력한다.
- http://localhost:8080/rqreport/

로그인 화면에서 임시 로그인 id/pw 를 입력한다.



● 서버 관리자 화면에서 설치 관리를 선택한 후, 관리 DB 생성을 클릭하면 관리 DB가 생성된다.



● 서버 관리자에서 REQUBE REPORT 가 샘플로 제공하는 WorldAV.mdb 와 Sample 폴더에 있는 문서를 등록하여 샘플 테스트를 할 수 있다

REQUBE REPORT Server 를 설치하면 서버가 설치된 폴더에는 다음과 같은 서브 폴더가 생긴다. 각 서브 폴더의 내용과 역할은 다음과 같다.

common	REQUBE REPORT Server 의 공통 모둘 폴더
css	Cascading style sheets 파일들
db	DB 관리 폴더
document	문서 리스트 및 문서관리 폴더
environment	환경 변수 폴더
img	서버 관리자에 사용되는 image 파일들
login	서버 관리자의 로그인 폴더
mdb	샘플 관리 DB(RQRepository.mdb)
mysingle	MySingle 연동을 위한 예제파일

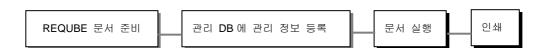


⊟ 🛅 Tomcat 5,0	sample	샘플 관리 DB(RQRepository.mdb)파일에 등록된 샘플문서
	setup\cab	뷰어 실행시 필요한 Cab(RQViewer.cab)파일
	tmp	임시 폴더
🫅 logs	usergroup	사용자/그룹 관리 폴더
ings ings ings ings ings ings ings ings	WEB-INF	Web 어플리케이션의 자바라이브러리 및 클래스파일

REQUBE REPORT Server의 기능을 한마디로 말하자면 REQUBE REPORT 문서를 웹 환경에서 실행하는 것이다. 그러므로 문서 실행을 위해 준비되어야 하는 것들과 그 과정을 알아본다는 것은 REQUBE REPORT Server를 이해하는 첫걸음이라 할 수 있다.

REQUBE REPORT Server 는 데이터베이스 연결, Query 실행과 사용자 인증, 서버의 정보 관리 기능을 수행한다.

클라이언트에서는 서버로부터 결과 데이터셋과 설계 문서를 받아 문서를 실행하며 서버 관리자 모듈을 이용하여 서버에 접속하여 서버 정보를 용이하게 관리할 수 있다.



REQUBE REPORT 문서(파일) 준비

REQUBE REPORT Designer 에서 작성하여 저장된 설계문서(*.rqx) 또는 실행문서(*.rqv)가 필요하다.

관리 DB 에 관리 정보 등록

문서 실행을 위해 필요한 관리 정보들이 REQUBE REPORT Server에 등록되어 있어야 한다. 문서를 실행 할 주체인 사용자, 실행 대상인 문서, 문서가 데이터를 가져올 원천인 데이터베이스의 정보가 우선적으로 등록되어 있어야 하며, 사용자의 데이터베이스 및 문서에 대한 권한도 등록되어 있어야 한다.

이러한 모든 관리 정보는 REQUBE REPORT Server를 설치시 함께 제공되는 서버 관리자 모듈을 통해 관리되며 관리 정보는 관리 정보 데이터베이스에 보관된다.

관리 정보 데이터베이스는 최초 접속 시에 [설치관리]-[관리 DB 생성]에서 생성한다. 관리 DB 가 생성되면 서버 관리자를 이용하여 문서, DB, 사용자 등 문서 실행에 필요한 정보를 등록한다.

🗿 문서 실행

문서 실행을 위해 관리 정보와 문서가 모두 등록되었다면 이제 문서 실행을 REQUBE REPORT Server에 요청할 수 있다. 이 때 서버는 관리 정보 연결와 데이터베이스 연결, Query 실행을 수행하며 실제 문서 실행은 클라이언트의 RQViewer에서 처리된다.

문서 실행 요청이 오면 다음의 과정을 거친다.

- □ 문서 실행을 요청한 사용자가 REQUBE REPORT Server 에 등록된 정당한 사용자인가를 인증한다.
- 🔼 인증된 사용자가 문서에 대한 실행권한이 있는지 인증한다.
- □ 데이터베이스와 연결하여 문서에 정의된 Query 를 실행한 후 결과 데이터셋과 함께 설계 문서를 클리이언트로 다운 로드한다.
- □ 클라이언트의 RQViewer의 Report Engine 에서 서버로부터 받은 설계 문서와 결과 데이터셋을 연결하여 문서를 실행한다.
- ☑ RQViewer가 실행 결과를 ActiveX로 결과 창에 보여준다.

🗿 이쇄

문서 실행 결과를 웹 브라우저 상에서 인쇄할 수 있다.(뷰어의 툴바에 있는 인쇄 버튼을 이용한다.)



1.

1.1

REQUBE REPORT Server(RQREPORT)는 사용자가 작성한 문서를 서버상에 등록하고, 이를 공유하여 여러 사용자가 각자의 목적에 맞게 문서를 실행할 수 있도록 해 준다. 그러기 위해서는 관리자라는 역할이 필요하고, 관리자는 REQUBE REPORT Server(RQREPORT)에서 서비스를 운영하는데 필요한 관리 정보에 대한 등록 및 유지 보수를 담당하는 역할을 하게 된다.

이 때, 관리 정보는 문서 관리, 사용자 관리, 데이터베이스, 설치 관리, 환경변수로 분류되어 관리된다.

문서 관리는 REQUBE REPORT Designer 에서 설계된 문서를 REQUBE REPORT Server 에 등록 관리하는 기능이며 이 때, 등록된 문서들을 기능 혹은 주제별로 분류하여 보관하는 폴더에 대한 관리도 함께 이루어진다.

.**사용자 관리**는 REQUBE REPORT Server 에 로그인하여 문서 실행을 하는 개개의 사용자를 등록 관리하는 한편 사용자들을 분류하여 그룹에 지정, 사용자 각각이 아닌 그룹 단위로 관리를 가능케 하여 편의성을 제공한다.

- □ 데이터베이스는 등록된 문서들의 실행 시 연동될 DB 정보를 등록 관리한다.
- □ 설치 관리에서는 뷰어파일 설치와 관리 DB 생성을 한다.
- 환경 변수 관리는 REQUBE REPORT Server 의 환경 변수를 수정/ 반영한다.
- 이러한 관리 정보는 서버의 관리 DB에 저장되며, 문서는 Repository 폴더에 저장된다.

1.2

REQUBE REPORT Server 의 관리 정보를 관리할 수 있는 서버 관리자로 로그인하는 방법은 다음과 같다.

웹 브라우저 상의 주소창에서 http://서버IP/rqreport/ 을 입력한다.

아이디와 비밀번호 부분에 관리자 아이디와 비밀번호를 입력하면 웹 서버 관리자로 로그인된다.



1.3

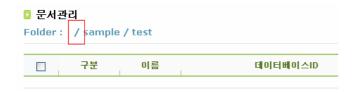
REQUBE REPORT 문서를 실행하기 위해서는 문서를 서버 상에 등록하는 것이 필요하다.

문서 관리에서는 문서의 등록 및 삭제, 문서를 실행하기 위한 정보 보관 설정을 할 수 있다.

이러한 문서 관리 기능를 용이하게 하여 주는 것이 폴더로 문서들을 기능 또는 주제별로 분류하여 한 폴더 밑에 넣어주면 문서들을 그룹핑하여 관리할 수 있다. 폴더 역시 등록 및 삭제가 가능하며 문서 권한 설정도 가능하다.

문서 관리에서는 REQUBE REPORT SERVER 의 환경 변수 중 Repository 에 지정된 폴더를 기본 폴더로 한다. 따라서 문서 관리에서 사용되는 폴더는 이 기본 폴더에서의 상대 위치이다.

1.3.1



<루트 폴더로 이동>

① 문서관리 화면에서 왼쪽 상단에 있는 네비게이션 바(현재 폴더 위치를 보여주는 곳)에서 커서를 원하 는 폴더 위치로 이동시켜 선택하면 현재 폴더의 위치 가 변경된다. 루트 폴더로 가고자 하는 경우는 맨 왼 쪽의 "/"를 선택하면 된다.





</Sample 폴더로 이동>

② 현재 폴더 아래에 있는 폴더와 문서 목록이 문서 관리 화면에 보여진다

1.3.2

① 폴더 이동으로 새 폴더를 만들 위치로 이동한다. 문서관 리 화면에서 오른쪽 하단 에 있는 <새 폴더 등록> 버튼을 클릭한다.



② 새 폴더 등록 대화상자의 폴더이름에 생성될 폴더의 이름을 입력한다



③ 새 폴더가 등록 되었음을 확인한다.



1.3.3



- 네비게이션바를 이용하여 삭제할 폴더로 이동한다.
- ② 문서관리 화면에서 삭제할 폴더를 선택한 후 왼쪽 하단에 있는 <삭제> 버튼을 클릭한다.
- ③ 삭제 확인 메시지가 나오면 <확인>버튼을 눌러 삭 제한다.

● 삭제하려는 폴더에 하위 폴더가 있거나 문서가 등록되어 있으면 삭제할 수 없다. 그러므로 폴더를 삭제하기 의해서는 하위 폴더와 등록된 문서를 먼저 삭제하여야 한다.



1.3.4



① 네비게이션바를 이용하여 문서를 등록할 폴더로 이 동한다.

② 문서관리 화면에서 오른쪽 하단에 있는 <새 문서 등록> 버튼을 클릭한다.

③ 새 문서 등록 대화상자에서 <추가> 버튼을 눌러 등록 할 문서를 찾아온 뒤 <확인>버튼을 누른다.



④ 찾아온 문서는 아래 문서 리스트에 나타난다. 여러 개의 문서를 반복적으로 찾아올 수 있다. 등록할 문서 를 모두 찾은 후 <확인>버튼을 누른다.



🧐 문서 관리 화면에서 등록된 문서들을 확인할 수 있다.



등록된 문서의 데이터베이스 설정을 해준다.

데이터베이스설정은 문서내 데이터베이스와 WAS에 등록된 데이터베이스와의 맵핑작업이며

문서를 선택하고 데이터베이스설정버튼을 누르면 대화창이 뜬다.

(저장된 JNDI 는 서버에 이미 저장되어 있는 JNDI 이므로 최초 데이이터베이스설정시 보이지 않는다.)





🕡 데이터베이스를 설정을 하고 나면 아래와 같이 등록된 🏻 🖲 문서 관리 화면에서 문서의 이름을 클릭하면 등록 데이터베이스 ID 가 보인다.



된 문서가 바로 실행된다.

	199	7-01-01 00 00 00	1997-02-01 00:00:00		1997-02-01 00:00:00		199	7-03-01 00:00:00	1997-04-01 00:00:00	
	주문건수	중주문급액	주문건수	충주문급역	주문건수	중주문금액	주문건수	중주문급역		
기양대리콥							1	2002102.000000		
김석대리장	1	472664,000000	2	795000,000000	3	5169612,000000	1	2666090,000000		
고전대리점	2	249500,000000	1	798000,000000			1	1841268,000000		
내지대리캠			1	865600,000000						
대치대리함	2	3460712,000000	2	1988153,000000	1	6322442,000000				
86033	1	1596386,000000	2	6738268,000000			2	3805432,000000		
마찬대리장			1	684758,000000			1	6860405,000000		
운전대전공										
21510101					2	3174888,000000	1	4567632,000000		
성내대리점	1	160800,000000					1	378000,000000		
용발대리점	3	6219607,000000	1	877338,000000	1	991453,000000	1	9786721,000000		
20002	2	7372483,000000	2	7442506,000000	2	3135530,000000	1	840636,000000		
西海田公吉	- 4	7498158,000000	1	169000,000000			3	2275673,000000		
상선대리장	2	2299720,000000	1	178000,000000			2	8145850,000000		
유용대의장	3	5744175,000000	1	299250,000000						
회목대리함	1	778206,000000	2	4116190,000000	1	792246,000000	2	11508498,00000		
축박대리장										
UN.	22	35852411,000000	17	24892063,000000	10	19586171,000000	17	54678307,00000		

1.3.5



- 💵 네비게이션바를 이용하여 문서가 있는 폴더로 이동한 다.
- 🥝 문서관리 화면에서 삭제할 문서를 선택한 후 왼쪽 하 단 에 있는 <삭제> 버튼을 클릭한다.
- ◎ 삭제 확인 메시지가 나오면 <확인>버튼을 눌러 삭제 하다.

1.3.6

서버에서 문서를 실행하기 위해서는 설계된 문서에 저장된 데이터베이스를 서버에 있는 데이터베이스로 설정 해 주어야 한다.



- 🕕 네비게이션바를 이용하여 데이터베이스를 설정할 문서 가 있는 폴더로 이동한다.
- 🥝 데이터베이스를 설정할 문서를 선택한다.

- 📵 다음과 같은 데이터베이스 설정 대화상자가 뜨면 🚨 저장된 JNDI 를 확인하고 변경해야 할 경우 변경할 문서 리스트에서 문서를 선택한다. 문서를 선택하면 문 JNDI 콤보 박스에서 현재 서버에 사용 가능한 JNDI 중 하



서 내 데이터베이스에 현재 문서에 정의된 데이터베이 나를 선택한 후 [확인] 버튼을 눌러 데이터베이스를 설정 스 정보가 나타난다.



한다.

⑤ [확인] 버튼을 누르면 변경된 JNDI를 확인할수 있다.



1.4

사용자관리에서는 새로운 사용자를 등록할 수 있다. 새 사용자 등록 버튼을 클릭하면 아래와 같은 화면이 나오며



사용자 등록시 유의사항은 다음과 같다.

사용자 ID:

- 길이는 5~20 자리 이어야한다.
- 한글과 특수문자는 사용 불가하며 첫글자는 영문으로 시작해야 한다.

암호 :

- 길이는 6~20 자리이어야하며 영문과 숫자의 혼용이어야한다.

권한 :

- 권한은 일반권한과 관리자권한이 있으며 일반은 문서관리 DB 만 사용할수 있으며 관리자는 모든 매뉴를 사용 할 수 있다.

1.5

데이터베이스를 선택하면 현재 WAS 에 등록된 데이터베이스의 현황를 보여준다. 여기서 주의할 점은 DB 관리에서



는 데이터베이스 현황만을 보여 줄 뿐이며 관리자가 임의로 수정 또는 삭제하는 것은 불가능하다.

보여지는 데이터베이스 정보는 서버 JNDI NAME, User Name, DB Server IP, SID, JDBC INFO 이다.



1.6

설치 관리에서는 뷰어 파일 설치에 관련된 기능과 관리 DB 생성 기능을 수행 할 수 있다.



❶ Viewer 현황에서는 현재 서버에 있는 RQViewer.cab 의 마지막 수정 일자와 크기 정보를 확인할 수 있다.

예) RQViewer.cab / lastModified : 070228 09:57:20 / size 2435687 /

🥝 **Viewer 파일 올리기**를 통하여 서버로 직접 RQViewer.cab 파일을 올릴 수 있다.

Viewer 파일 올리기를 선택하여 viewer 등록 대화 상자에서 서버로 올릴 뷰어의 위치를 <찾아보기>버튼을 이용하여 지정한다.<확인> 버튼을 누르면 서버로 선택된 뷰어 파일이 업 로드 된다.





새로운 RQViewer.cab 파일을 올리면 Viewer 현황에 이전 버전과 새로 등록된 뷰어파일을 확인할 수 있다

Viewer 현황 RQViewer.cab / lastModified : 070123 20:19:16 / size : 2435687 / RQViewer.cab.OLD / lastModified : 070119 09:43:02 / size : 2435687 /

- ③ Viewer 설치를 클릭하면 등록되어 있는 RQViewer.cab 파일을 수동으로 설치하게 된다.
- Viewer 파일 rollback 을 클릭하면 뷰어를 이전 버전으로 복구 시킨다.
- ⑤ 관리 DB 생성을 클릭하여 REQUBE REPORT Server 을 운영할 때 필요한 관리 DB 를 설치할수 있다. (일반적으로 관리 DB 는 REQUBE REPORT Server 를 최초설치 한후 한번만 실행한다.)

1.7

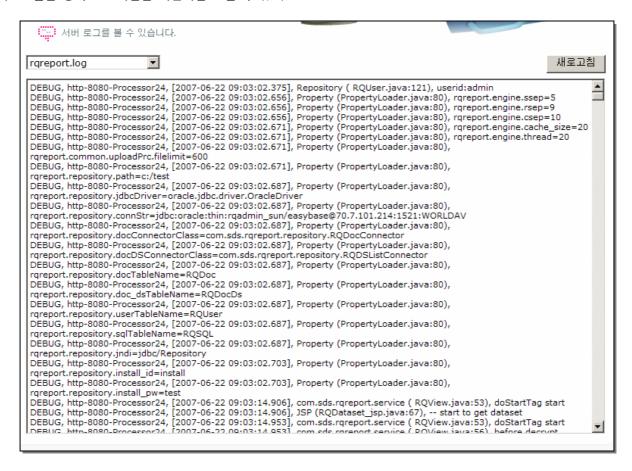
환경변수 관리에서는 REQUBE REPORT Server에 지정된 환경 변수의 값을 변경할 수 있다. 환경 변수 관리 화면에서 기본값과 현재 설정된 적용값을 확인한 후 적용값에서 원하는 값으로 변경한 뒤 <반영>버튼을 눌러 변경된 값을 시스템에 반영한다.



1.8

로그모니터는 각 레벨별로 DEBUG(디버그), INFO(정보), WARN(경고), ERROR(에러), FATAL(치명) 로 표시가 되며 서버 로그파일에 직접 접근하여 보지않고도 로그파일을 웹상에서 날짜별로 볼 수 있도록 하는 기능을 제공한다.

리스트박스를 클릭하면 날짜별 로그파일을 볼수 있으며 해당 파일을 선택하면 선택한날짜의 로그파일을 볼수 있으며, 새로고침을 통해 로그파일을 최신파일로 볼 수 있다.



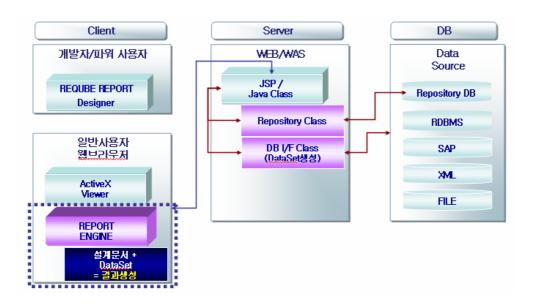
2.

REQUBE REPORT 문서를 디자이너를 이용하여 설계한 다음, 서버 업로드 기능이나 서버 관리자를 통하여 REQUBE REPORT Server에 등록하였다면 문서를 실행할 수 있는 모든 준비가 되었다고 할 수 있다.

그럼, 이제 REOUBE REPORT 문서가 어떠한 절차와 방법으로 웹 환경에서 실행되는지를 살펴 보도록 하겠다.

2.1

다음은 REQUBE REPORT Server 의 제품 구성도로 클라이언트의 뷰어, 서버의 JSP/Servlet, Report server 와 DB Engine 으로 구성되어 있다. 클라이언트의 뷰어는 문서의 실행, JSP/Servlet 은 JSP의 실행, Report Server 는 문서의 관리, DB Engine 은 데이터베이스 연결 기능을 수행하고 있다.



REQUBE REPORT Server 에서 문서 실행 과정과 과정에서의 주체를 살펴 보면 다음과 같다.

- ❶ 클라이언트에서 서버로 JSP를 이용하여 문서 실행 요청을 한다.
- ② 서버에 있는 Reporting Server 의 Repository Manager 는 서버에 정의된 Repository 폴더에서 해당 문서를 찾아 클라이언트로 보내준다.
- ③ 클라이언트의 뷰어는 받아온 문서에서 정의된 Query 문을 추출하여 웹 서버에 Query 실행을 요청한다.
- 서버의 DB Engine 은 서버의 데이터베이스를 JNDI를 통해 연결하고 Query를 실행한다. 이후 실행 결과 데이터 셋을 클라이언트로 보낸다.
- ⑤ 클라이언트의 뷰어는 서버로부터 받은 결과 데이터셋과 문서를 연결하여 문서 실행을 한 후 실행 결과를 클라이언트 화면 또는 프린터로 출력한다.
- 이 과정 중에서 가장 큰 주체가 되는 것은 클라이언트의 뷰어임을 알 수 있다. 즉, 서버의 주 기능은 관리 정보(문 서포함)의 보관과 데이터베이스 연결이며 실제 문서 실행은 클라이언트에서 처리된다. 따라서 서버의 부하를 줄이 는 Thin server-Rich Client 의 구조를 갖는다.

2.2 JSP

다음은 문서 실행과 관련하여 REQUBE REPORT Server 설치 시 함께 제공되는 JSP이다. 이 JSP를 바로 이용하거나 필요 부분을 수정하여 문서 실행에 활용할 수 있다.(REQUBE REPORT Server 가 설치된 폴더에 위치하고 있다.)

- [WebRoot]/document/getreport.jsp ⇒ 문서를 다운로드 받는다. Repository 로부터 문서를 찾아서 클라이언트로 보 낸다.
- □ [WebRoot]/rqviewer.jsp => 문서 실행에 관련된 html 을 생성한다.
- [WebRoot]/RQDataset.jsp ⇒실행요청을 받아서 결과셋을 리턴해 준다.

Repository Manager 는 WAR 패키징 안에 포함되며 관리 DB 를 사용하지 않는 경우는 문서를 WebServer 를 통해 직접 다운로드 할 수 있게 WebServer 를 설정하면 된다. 즉, rqviewer.jsp 에서 문서를 가져오는 부분을 변경해서 직접 다운로드하는 URL로 변경하면 된다. → Ex)http://[서버]/[경로]/문서이름.rqx로 문서를 다운로드한다

위와 같이 사용하는 경우에는 서버 관리자는 사용할 수 없으며 직접 문서를 올리고 디렉토리를 구성해야 한다. 그리고 문서를 직접 URL 입력을 통하여 다운로드 할 수 있기 때문에 보안에 취약하다.

2.3 rqviewer.jsp

rqviewer.jsp 에 문서명(doc) 인자를 넘기면 문서가 실행 된다.

rqviewer.jsp 에 문서명을 넣어서 실행하면 javascript 가 포함된 html 이 생성되는데 그 html 에서 뷰어를 로딩하며 해당 스크립트를 통해서 ActiveX 뷰어(RQViewer)로 인자값을 전달하고 HTML 이 로딩시 Javascript 함수 OnRun 을 호출하여 실행하게 된다. (Viewer API 는 3.4 - ROViewer API 부분을 참조한다.)

rqviewer.jsp 는 다음과 같은 5개 부분으로 나뉘어져 있다.

- 🔼 RQViewer 개체의 OpenReport를 사용한 문서열기 부분
- CreateConnectionControl 을 사용하여 connection 개체 만들기 부분.
- connection 개체의 설정 부분
- 데이터셋 설정
- 얼어진 데이터 셋을 통한 문서 실행 부분

rqviewer.jsp 에는 인증 부분이 빠져 있으므로 인증을 추가하기 위해서는 JSP 에서 인증을 추가하여야 한다. 문서를 실행하는 jndi 명을 parameter 로 받게 되어 있으나 RepositoryManager 를 사용하는 경우에는 문서가 실행하는데 사용하는 jndi 명을 등록된 서버에서 얻는다.

다음은 실제 rqviewer.jsp 이다. 중간의 설명 부분을 참조하여 살펴보도록 한다.

```
<@@ page contentType="text/html;charset=EUC-KR"%>
<%
         String path = request.getContextPath();
        String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
         String Im_doc = request.getParameter("doc");
         String doc = new String(Im_doc.getBytes("8859_1"),"KSC5601");
         String Im_runvar = request.getParameter("runvar");
         String runvar;
        if (Im_runvar != null)
                  runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
        else
                 runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
function OnRun(docName,runvar)
{
        var oReport;
        var oSQL;
        var oConnection;
        var OConnObj;
        var oDataSet;
        var oDataSetCtrl;
        //---OpenReport 를 사용하여 문서열기------
        if (docName == 'null')
                  alert("DocName is Null");
                  return;
        }
```

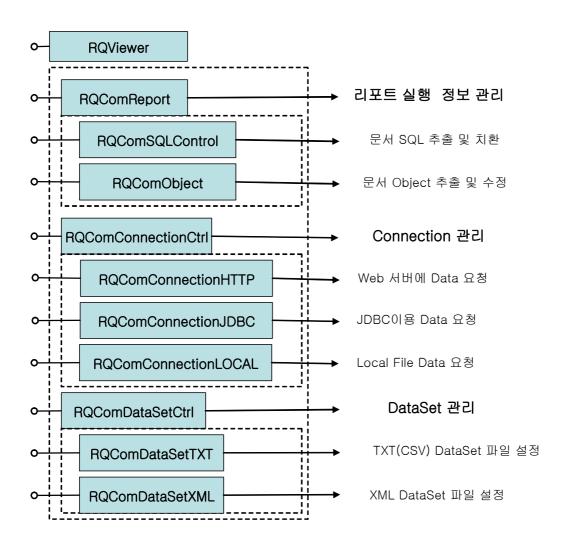
```
else
{
       try
       {
           oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
       } catch(err)
       {
               return;
       }
}
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//---CreateConnectionControl 을 사용하여 connection 개체 만들기-----
oConnection = RQViewer.CreateConnectionControl();
//---Connection 개체의 설정-----
OConnObj = oConnection.AddConnection("http.post");
OConnObj.AddParameter("action", "getRsDB");
OConnObj. Add Parameter ("runvar", oReport. GetRuntime Variable ());\\
OConnObj.AddParameter("doc",docName);
OConnObj.SetQueryCount(oSQL.GetQueryCount());
OConnObj.SetPath("<%=basePath%>RQDataset.jsp");
//---DataSet 을 설정-----
oDataSetCtrl = RQViewer.CreateDataSetControl();
for (i = 0; i< oConnection.GetCount();i++)
       oConnObj = oConnection.GetConnection(i);
       for (j = 0; j < oConnObj.GetCount(); j++)
```

```
{
                         oDataSet = oDataSetCtrl.AddDataSet("txt");
                         oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j));
                         oDataSet.SetEncoding(oConnObj.GetEncoding());
                         oDataSet.SetSeperator(oSQL.GetDefaultRowSeperator(),oSQL.GetDefaultColSeperator());
                }
        }
        //---문서실행-----
        RQViewer.Run();
-->
</SCRIPT>
<STYLE>
        DIV { padding-top: 0px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px;}
</STYLE>
</HEAD>
<BODY topmargin="0" leftmargin="0" onload="OnRun('<%=doc%>','<%=runvar%>')">
<script language="javaScript" SRC="./setup/rqviewer.js"></script>
</BODY>
</HTML>
```

3. RQViewer

3.1

RQViewer 는 리포트 설계 문서 정보를 실행/관리하는 Report Control, 결과 Data 를 얻어오는 Connection 을 관리하는 Connection Control 및 DataSet 들을 관리하는 DataSetCtrl 로 크게 3 부분으로 구성된다.



3.2 API

3.2.1 (RQViewerCtrl)

HTTP 호출 JSP 예제 (rqviewer_http.jsp)

<%@ page contentType="text/html;charset=EUC-KR"%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";</pre>

```
String Im_doc = request.getParameter("doc");
        String doc = new String(Im_doc.getBytes("8859_1"),"KSC5601");
        String Im_runvar = request.getParameter("runvar");
        String runvar;
        if (Im_runvar != null)
                 runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
        else
                 runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
function OnRun(docName,runvar)
{
        var oReport;
        var oSQL;
        var oConnection;
        var OConnObj;
        var oDataSet;
        var oDataSetCtrl;
        //---Viewer 속성(Property) 세팅 부분-----
        //RQViewer.BackColor = 255;
        //RQViewer.ToolBarVisible = false;
        //RQViewer.StatusBarVisible = false;
        //RQViewer.ShowProgressDialog = false;
        //RQViewer.EmptyDataCheckOption = "part"; //"all" or "part" or "Query1|Query2"
        //---OpenReport 를 사용하여 문서열기-----
        if (docName == 'null')
```

```
alert("DocName is Null");
       return;
}
else
{
       try
       {
          oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
      } catch(err)
       {
              return;
      }
}
//---문서연결을 위한 BaseURL 설정------
RQViewer.SetBaseURL("<%=basePath%>");
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//---CreateConnectionControl 을 사용하여 Connection 개체 만들기------
oConnection = RQViewer.CreateConnectionControl();
//---Connection 개체의 설정-----
OConnObj = oConnection.AddConnection("http.post");
OConnObj.AddParameter("action", "getRsDB");
OConnObj. Add Parameter ("runvar", oReport. GetRuntime Variable ());\\
OConnObj.AddParameter("doc",docName);
OConnObj.SetQueryCount(oSQL.GetQueryCount());
OConnObj.SetPath("<%=basePath%>RQDataset.jsp");
//---DataSet 을 설정-----
```

```
oDataSetCtrl = RQViewer.CreateDataSetControl();
        for (i = 0; i< oConnection.GetCount();i++)
                oConnObj = oConnection.GetConnection(i);
                for (j = 0; j < oConnObj.GetCount(); j++)
                {
                        oDataSet = oDataSetCtrl.AddDataSet("txt");
                        oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j));
                        oDataSet.SetEncoding(oConnObj.GetEncoding());
                        oDataSet.SetSeperator(oSQL.GetDefaultRowSeperator(),oSQL.GetDefaultColSeperator());
                }
        }
        //---문서실행-----
        RQViewer.Run();
//[----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
function RQ_HyperLinkSummit()
{
        RQ_HyperLink.submit();
function RQ_SetTarget(val)
{
        RQ_HyperLink.target = val;
}
function RQ_SetAction(val)
{
        RQ_HyperLink.action = val;
```

```
function RQ_AddValue(name,val)
{
        if(document.getElementById(name)!= null)
                 document.getElementById(name).value = val;
        }
        else{
                 elem = document.createElement("input");
                 elem.type="hidden";
                 elem.name=name;
                 elem.id=name;
                 elem.value=val;
                 RQ_HyperLink.appendChild(elem);
        }
}
function RQ_ExcuteHyperLink(url,target,option)
{
        window.open(url,target,option);
//]----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
</SCRIPT>
<!-- 이벤트 설정 -->
<script language="JavaScript" for="RQViewer" event="EmptyResultData">
        alert("EmptyResultData");
</script>
<script language="JavaScript" for="RQViewer" event="EndRunReport">
        alert("EndRunReport");
</script>
<script language="JavaScript" for="RQViewer" event="Refresh">
        alert("Refresh");
```

RQViewerCtrl - HTML <OBJECT> 코드 (./setup/rqviewer.js)

document.write('<object id="RQViewer" classid="clsid:94542C25-D867-4C5D-BD5C-FFAFD9916C0B" codebase="/rqreport/setup/cab/RQViewer.cab#version=1,0,7,0712" WIDTH="100%" HEIGHT="100%">'); document.write('</object>');

-RQViewerCtrl rqviewer.js RQViewer

OpenReport

OpenReport 함수는 로컬 또는 웹 서버에 있는 REQUBE 파일(.rqd)을 뷰어로 다운받아 오픈 한 후 리포트 컨트롤 (RQReport)를 생성하여, 반환하는 함수이다. 파일이 잘못되었거나, 경로에 파일이 없는 경우 NULL 객체를 반환한다.

□ 함수 정의 및 인자



[RQReport 컨트롤 객체] = OpenReport([파일경로])
RQReport Ctrl 객체를 생성하여 리턴한다
[파일경로] - REQUBE 파일의 URL 및 로컬파일 경로를 설정

■ Web 서버에 있는 보고서 설계 문서를 오픈 하는 경우의 예제

var oReport = RQViewer.OpenReport("http://www.reqube.com/sample.rqd");

웹 서버의 주소 변경 시 소스 수정의 번거로움을 피하기 위해 다음과 같이 location.host 객체 프로퍼티를 사용한다. => oReport = RQViewer.OpenReport("http://" + location.host + "/sample.rqd");

□ 로컬에 있는 보고서 설계 문서를 오픈 하는 경우의 예제

var oReport = RQViewer.OpenReport("c:/sample.rqd");

OpenReport 호출 시 로컬 경로에 있는 보고서 설계파일을 오픈하고, 리포트 컨트롤 객체를 생성하여 리턴한다.

CreateConnectionControl

ConnectionControl 을 생성하여 반환하고, 이 ConnectionControl 에 여러가지 종류의 Connection 을 추가 할 수 있고, 추가한 순서대로 Connection 을 생성하여 요청을 실행하고, Connection 을 관리하는 역할을 한다. (RQConnection 항목참고)

□ 함수 정의 및 인자

[RQConnectionCtrl 객체] = CreateConnectionControl()

Connection Ctrl 을 생성하여 리턴한다.

[RQConnectionCtrl 객체] = CreateConnectionControl()
Connection Ctrl 을 생성하여 리턴한다

RQConnectionCtrl 은 Send 나 Run 함수 실행 시, 추가된 Connection 을 순서대로 실행하여, 결과 DataSet 을 받아오는 명령을 수행하게 된다.

Run



Run 함수는 리포트를 실행하는 함수로, Thread 를 이용하여 결과 Data Set 을 받아 리포트를 실행하게 된다.

□ 함수 정의 및 인자

Run()
리포트 실행을 Thread 를 이용하여 실행한다.
없음

Create Data Set Control

RQDataSetControl 객체를 생성하여 반환하고, 여러 가지 종류의 DataSet 을 추가하고, 관리하게 한다. (RQDataSetControl 부분 참고)

▶ 함수 정의 및 인자

[RQDataSetCtrl 객체] = CreateDataSetControl()
RQDataSetCtrl 객체를 생성하여 반환 한다
없음

GetReport

RQReport 컨트롤 객체를 반환 한다. RQReport 객체가 OpenReport 를 통하여 생성되지 않았으면 NULL 값을 리턴한다.

□ 함수 정의 및 인자

[RQReport 객체] = GetReport()
RQReport 객체를 반환 한다
없음

GetConnectionControl 함수

RQConnectionCtrl 객체를 반환한다. CreateConnectionControl 을 이용하여 객체가 생성되어있지 않은 경우 NULL 값을 리턴한다.



□ 함수 정의 및 인자

[RQConnectionCtrl 객체] = GetConnectionControl()
RQConnectionCtrl 객체를 반환 한다.
없음

GetDataSetControl 함수

DataSetControl 객체를 반환한다. CreateDataSetControl 을 이용하여 객체가 생성되어 있지 않으면 NULL 값을 반환한다.

▷ 함수 정의 및 인자

[DataSetCtrl 객체] = GetDataSetControl()
DataSetCtrl 객체를 반환한다.
없음

CloseReport

CloseReport 함수는 실행된 결과 리포트를 닫고, 뷰어를 실행 이전으로 초기화 한다.

▶ 함수 정의 및 인자

CloseReport()
실행한 리포트를 닫고 뷰어를 초기화 한다.
없음

Stop

Stop 함수는 실행 중인 리포트를 중지한다. 실행된 부분까지 결과를 보여준다.

▶ 함수 정의 및 인자



Stop()
리포트 실행을 중지한다.
없음

Refresh

문서를 재실행 하여 결과를 다시 생성한다. Refresh 함수 호출 이전에 설정 값들을 변경하면 변경된 값을 문서를 실행하여 결과를 생성한다.

▶ 함수 정의 및 인자

Refresh()
리포트를 설정 값으로 다시 실행한다
없음

MoveFirtstPage

결과 보고서의 첫 페이지로 이동하는 함수이다.

□ 함수 정의 및 인자

MoveFirstPage()
결과 보고서의 첫 페이지로 이동한다.
없음

▶ 사용예제

MoveLastPage

결과 보고서의 마지막 페이지로 이동한다.

▶ 함수 정의 및 인자

MoveLastPage()
결과 보고서의 마지막 페이지로 이동한다.
없음

▶ 사용예제

MoveNextPage

결과 보고서의 현재 페이지의 다음 페이지로 이동한다.

□ 함수 정의 및 인자

MoveNextPage()
결과 보고서의 현재 페이지의 다음 페이지로 이동한다.
없음

▶ 사용예제

<SCRIPT LANGUAGE="javascript">

MovePrevPage

결과 보고서의 현재 페이지의 이전 페이지로 이동한다.

▶ 함수 정의 및 인자

MovePrevPage()
결과 보고서의 현재 페이지의 이전 페이지로 이동한다
없음

▶ 사용예제

MovePage

지정된 페이지로 이동하는 함수이다.

□ 함수 정의 및 인자



MovePage([페이지 번호])
지정된 페이지로 이동하는 함수이다.
[페이지 번호] - 이동하고자 하는 페이지 번호

▶ 사용예제

ZoomOut

화면을 25%의 비율로 축소한다.

□ 함수 정의 및 인자

ZoomOut()
한번에 화면을 25%의 비율로 축소한다
없음

🗅 사용예제

ZoomIn

화면을 25%의 비율로 확대한다.

▶ 함수 정의 및 인자

Zoomln()
화면을 한번에 25%의 비율로 확대한다.
없음

사용예제

Zoom

화면을 지정한 비율로 확대 또는 축소한다.

□ 함수 정의 및 인자

Zoom([비율])
화면을 지정한 비율로 확대 또는 축소한다.
[비율] - 확대/축소 할 비율(%)

사용예제

<SCRIPT LANGUAGE="javascript">

```
function zoom(zoom)

{

RQViewer.Zoom(zoom);
}

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>
```

ShowToolBarButton

지정된 툴 바의 버튼을 숨기거나 보이도록 한다.

(버튼 추가에 따라 번호가 달라질수 있다. 번호는 툴바의 버튼의 순서를 지칭한다.)

▶ 함수 정의 및 인자

| ShowToolBarButton([버튼 번호],[보이기 여부]) |
|-------------------------------------|
| 지정한 툴 바의 버튼을 숨기거나 보이도록 한다. |

```
[버튼 번호] -
            0. 저장 버튼
            1. 구분 바
            2. 인쇄 버튼
            3. 미리 보기 버튼
            4. 구분 바
            5. 정지 버튼
            6. 새로 고침 버튼
            7. 구분 바
            8. 처음 페이지 이동 버튼
            9. 이전 페이지 이동 버튼
            10. 페이지 번호 콤보박스
            11. 다음 페이지 이동 버튼
            12. 마지막 페이지 이동 버튼
            13. 구분 바
            14. 확대/축소 비율 텍스트 박스
            15. 구분 바
            16. 엑셀로 보내기
            17. PDF 로 보내기
            18. 구분 바
            19. 등록 정보 버튼
```

▶ 사용예제

[보이기 여부] - true : 보이기 / false : 숨기기

```
// 정지 버튼 숨기기
       RQViewer.ShowToolBarButton(5, false);
       // 새로 고침 버튼 숨기기
       RQViewer.ShowToolBarButton(6, false);
       // 처음 페이지 이동 버튼 숨기기
       RQViewer.ShowToolBarButton(8, false);
       // 이전 페이지 이동 버튼 숨기기
       RQViewer.ShowToolBarButton(9, false);
       // 페이지 번호 콤보박스 숨기기
       RQViewer.ShowToolBarButton(10, false);
       // 다음 페이지 이동 버튼 숨기기
       RQViewer.ShowToolBarButton(11, false);
       // 마지막 페이지 이동 버튼 숨기기
       RQViewer.ShowToolBarButton(12, false);
       // 확대/축소 비율 텍스트 박스 숨기기
       RQViewer.ShowToolBarButton(13, false);
       // 엑셀로 보내기 버튼 숨기기
       RQViewer.ShowToolBarButton(16, false);
       // PDF 로 보내기 버튼 숨기기
       RQViewer.ShowToolBarButton(17, false);
       // 등록 정보 버튼 숨기기
       RQViewer.ShowToolBarButton(19, false);
</SCRIPT>
<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>
```

Print

결과 화면을 인쇄 한다.

▶ 함수 정의 및 인자

Print([인쇄대화상자보기여부],[시작페이지],[마지막페이지],[인쇄부수])



| 결과 화면을 인쇄한다. |
|---|
| [인쇄대화상자보기여부] - 인쇄대화상자 보기여부 (true/false) |
| [시작페이지] — 시작페이지 (추후 개발 적용 예정,0 값 입력) |
| [마지막페잊] — 마지막페이지 (추후 개발 적용 예정,0 값 입력) |
| [인쇄부수] — 인쇄 부수 (추후 개발 적용 예정,0 값 입력) |

▶ 사용예제

PrintDialog

결과 화면을 인쇄 대화상자를 이용하여 인쇄한다.

▶ 함수 정의 및 인자

| PrintAsDialog([옵션]) |
|----------------------------|
| 결과 화면을 인쇄 대화상자를 이용하여 인쇄한다. |
| [옵션] - (추후 개발 예정,"" 값 입력) |

사용예제

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

PrintPreview

결과 화면의 인쇄 미리보기를 실행한다.

▶ 함수 정의 및 인자

| PrintPreview() |
|-----------------------|
| 결과 화면의 인쇄 미리보기를 실행한다. |
| 없음 |

▶ 사용예제

SaveAsDialog

결과를 저장 대화상자를 이용하여 저장한다.

□ 함수 정의 및 인자

| SaveAsDialog([옵션]) |
|---|
| 결과를 저장 대화상자를 이용하여 저장한다. |
| [옵션] - 저장 대화 상자 옵션 (추후 개발 적용 예정, "" 값 입력) |

▶ 사용예제

SaveAs

결과를 저장 대화상자 없이 바로 인자를 이용하여 저장한다..

□ 함수 정의 및 인자

| SaveAs([타입],[저장위치],[시작페이지],[마지막페이지],[옵션]) |
|--|
| 결과를 저장 대화상자 없이 바로 인자를 이용하여 저장한다. |
| [타입] - 저장문서 타입 (rqv - REQUBE 결과, xls - 엑셀, pdf - PDF, rtf - Word 파일, |
| [저장위치] — 문서를 저장할 위치 |
| [시작페이지] — (추후 개발 예정, 0 값 입력) |
| [마지막페이지] — (추후 개발 예정, 0 값 입력) |
| [옵션] — (추후 개발 예정, "" 값 입력) |

🗅 사용예제

SendToExcel



리포트 실행 결과를 엑셀로 변환한다.

□ 함수 정의 및 인자

| SendToExcel([저장위치],[포맷여부],[포맷제거옵션],[엑셀로딩여부]) |
|--|
| 리포트 실행 결과를 엑셀로 변환한다. |
| [저장위치] — 문서를 저장할 위치, "" 값일 경우 Temp 폴더에 저장한다. |
| [포맷여부] - 포맷 적용 여부 (true/false) |
| [포맷제거여부] — 포맷 부분 제거 옵션 |
| [엑셀로딩여부] — 엑셀 화면 로딩 여부 (true/false) |

▶ 사용예제

SendToPDF

리포트 실행 결과를 PDF로 변환한다.

▶ 함수 정의 및 인자

| SendToPDF([저장위치],[로딩여부]) |
|--|
| 리포트 실행 결과를 PDF로 변환한다. |
| [저장위치] — 문서를 저장할 위치, "" 값일 경우 Temp 폴더에 저장한다. |
| [로딩여부] – PDF 뷰어 로딩 여부 (true/false) |

▶ 사용예제

SetBaseURL

REQUBE 문서연결시 사용할 BaseURL을 설정한다..

□ 함수 정의 및 인자

| SetBaseURL([Base URL]) |
|--|
| REQUBE 문서연결시 사용할 BaseURL을 설정한다. |
| [Base URL] - REQUBE 문서연결시 사용할 BaseURL을 설정한다. |
| 문서 등록된 상대 폴더 위치의 앞부분의 URL을 설정한다. |

Cross Tab Page Break

REQUBE 통계문서 결과에 PageBreak 를 넣는다..

□ 함수 정의 및 인자

| CrossTabPageBreak() |
|-----------------------------------|
| REQUBE 통계문서 결과에 Page Break 를 넣는다. |
| 없음 |

▶ 사용예제

<SCRIPT LANGUAGE="javascript">

function crosstabpagebreak()

GetDocType

REQUBE 설계문서의 종류를 리턴한다.

▶ 함수 정의 및 인자

| GetDocType() |
|---|
| REQUBE 설계문서의 종류를 리턴한다.(양식형 보고서 = 1, 집계형 보고서 = 2) |
| 없음 |

GetTotalPageNo

REQUBE 양식형 보고서 문서의 전체 결과 페이지 값을 리턴한다.

□ 함수 정의 및 인자

| GetTotalPageNo() |
|---------------------------------------|
| REQUBE 양식형 보고서 문서의 전체 결과 페이지 값을 리턴한다. |
| 없음 |

▶ 사용예제

```
<SCRIPT LANGUAGE="javascript">
function totalpageno()
{
    return RQViewer.GetTotalPageNo();
}
</SCRIPT>
```

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

GetCurrentPageNo

REQUBE 양식형 보고서 문서의 현재 결과 페이지 값을 리턴한다.

▶ 함수 정의 및 인자

| GetCurrentPageNo() |
|---------------------------------------|
| REQUBE 양식형 보고서 문서의 현재 결과 페이지 값을 리턴한다. |
| 없음 |

▶ 사용예제

```
<SCRIPT LANGUAGE="javascript">
function currentpageno()
{
    return RQViewer.GetCurrentPageNo();
}
</SCRIPT>
<SCRIPT>
<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT></script>
```

BackColor

뷰어의 문서 실행 이전 바탕색을 설정하는 속성이다.

▶ 속성 정의

BackColor = [색상]

▶ 사용 예제 - JavaScript 사용

```
REQUBE REPORT Server 6.0
```

JavaScript 에서 색상 값 설정을 위하여 색상 값을 반환하는 RGB 함수를 만들어 사용한다.

▶ 사용 예제 - Object Loading Code 에서 사용 (./setup/rqviewer.js 파일)

```
document.write('<object id="RQViewer" classid="clsid:94542C25-D867-4C5D-BD5C-FFAFD9916C0B" width="100%" height="100%">');

document.write('<param name="BackColor" value="255">');

document.write('</object>');
```

Object Loading 시에 설정할 경우 RGB 색상 값을 설정한다.

ToolBarVisible

뷰어의 툴 바의 보이기 여부를 설정한다.

□ 속성 정의

ToolBarVisible = [true/false]

StatusBarVisible



뷰어의 상태 바의 보이기 여부를 설정한다.

□ 속성 정의

StatusBarVisible = [true/false]

ShowProgressDialog

보고서 실행 중 메시지 박스 보이기 여부를 설정한다.

□ 속성 정의

ShowProgressDialog = [true/false]

EmptyDataCheckOption

보고서 실행 중 Empty Data 판정 여부를 선택한다.

□ 속성 정의

EmptyDataCheckOption = "all" or "part" or "쿼리명 1|쿼리명 2"

▶ 사용 예제 - JavaScript 사용

// 뷰어에서 모든 쿼리의 데이터가 없을 경우 Empty Data 처리

RQViewer. EmptyDataCheckOption = "all"

// 뷰어에서 모든 쿼리의 데이터중 하나의 데이터 결과라도 없을 경우 Empty Data 처리

RQViewer. EmptyDataCheckOption = "part"

// 뷰어에서 특정 쿼리 Query1, Query3 의 쿼리의 결과 중 하나라도 없을 경우 Empty Data 처리

RQViewer.EmptyDataCheckOption = "Query1|Query3"

EndRunReport 이벤트

EndRunReport 이벤트는 리포트 실행이 완료된 후에 발생한다. 리포트 생성이 끝난 후 추가적인 작업을 할 경우 사용할 수 있다.

□ 이벤트 정의

EndRunReport

▶ 사용 예제

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="EndRunReport">

alert("리포트 실행이 완료되었습니다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

EmptyResultData 이벤트

EmptyResultData 이벤트는 리포트 실행이 완료된 후에 결과 데이터가 없을 때 발생한다. 리포트 생성이 끝난 후 결과데이터가 없을 때 추가적인 작업을 할 경우 사용할 수 있다.

□ 이벤트 정의

EmptyResultData

▶ 사용 예제

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" EmptyResultData">

alert("리포트 결과가 없습니다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

Refresh 이벤트

Refresh 이벤트는 리포트의 재실행 명령이 실행된 경우 발생하는 이벤트이다.

▶ 이벤트 정의

Refresh(Cancel)

Cancel - True: 재 실행 수행 , False: 재 실행 취소

▶ 사용 예제 - JavaScript 사용 예제

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="Refresh">

alert("리포트가 재실행되었습니다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 Refresh 가 실행 될 경우 Refresh 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT="Refresh(Cancel)">

Dim MyVar

MyVar = MsgBox("재실행하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 Refresh 실행을 취소 할 수 있다.

ButtonSaveAsClickBefore 이벤트

ButtonSaveAsClickBefore 이벤트는 문서 저장 버튼이 클릭 되었을 때 저장 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

Button Save As Click Before (Cancel)

Cancel - True: 파일 저장 대화상자가 나타남 (실행), False: 파일 저장 대화상자가 나타나지 않음 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonSaveAsClickBefore">

alert("저장버튼이 Click 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonSaveAsClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonSaveAsClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("저장하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 SaveAs 실행을 취소 할 수 있다.

</SCRIPT>

REQUBE REPORT Server 6.0

ButtonSaveAsClickAfter 이벤트

ButtonSaveAsClickAfter 이벤트는 파일 저장 버튼이 클릭 되어 저장 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

Button Save As Click After

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonSaveAsClickAfter">

alert("저장버튼이 Click 이후 이벤트이다.");

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonPrintClickBefore 이벤트

ButtonPrintClickBefore 이벤트는 문서 인쇄 버튼이 클릭 되었을 때 인쇄 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

Button Print Click Before (Cancel)

Cancel - True: 문서 인쇄 대화상자가 나타남 (실행), False: 문서 인쇄 대화상자가 나타나지 않음 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonPrintClickBefore">

alert("인쇄 버튼이 Click 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonPrintClickBefore 이벤트만 발생한다.

▶ 사용 에제 - VBScript 사용 예제

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonPrintClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("인쇄하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 인쇄 실행을 취소 할 수 있다.

ButtonPrintClickAfter 이벤트

ButtonPrintClickAfter 이벤트는 인쇄 버튼이 클릭 되어 인쇄 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

ButtonPrintClickAfter

사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonPrintClickAfter">

alert("인쇄버튼이 Click 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonRefreshClickBefore 이벤트

ButtonRefreshClickBefore 이벤트는 새로 고침 버튼이 클릭 되었을 때 새로 고침 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

ButtonRefreshClickBefore (Cancel)

Cancel - True: 새로 고침 명령을 실행함 (실행), False: 새로 고침 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonRefreshClickBefore">

alert("새로 고침 버튼이 Click 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonRefreshClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonRefreshClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("새로 고침하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 새로 고침 실행을 취소 할 수 있다.

ButtonRefreshClickAfter 이벤트

ButtonRefreshClickAfter 이벤트는 새로 고침 버튼이 클릭 되어 새로 고침 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

ButtonRefreshClickAfter

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonRefreshClickAfter"> alert("저장버튼이 Click 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonStopClickBefore 이벤트

ButtonStopClickBefore 이벤트는 정지 버튼이 클릭 되었을 때 정지 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

ButtonStopClickBefore (Cancel)

Cancel - True: 정지 명령을 실행함 (실행), False: 정지 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonStopClickBefore"> alert("정지 버튼이 Click 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonStopClickBefore 이벤트만 발생한다.

■ 사용 예제 - VBScript 사용

<HTML>

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonStopClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("정지 하시겠습니까?",vbOKCancel)

삼성 SDS

REQUBE REPORT Server 6.0

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 정지 명령 실행을 취소 할 수 있다.

ButtonStopClickAfter 이벤트

ButtonStopClickAfter 이벤트는 정지 버튼이 클릭 되어 정지 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

ButtonStopClickAfter

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonStopClickAfter">

alert("정지 버튼이 Click 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonZoomOutClickBefore 이벤트

ButtonZoomOutClickBefore 이벤트는 축소 API 가 호출되었을 때 축소 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

Button Zoom Out Click Before (Cancel)

Cancel - True: 축소 명령을 실행함 (실행), False: 축소 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonZoomOutClickBefore">

alert("축소 API 호출 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonZoomOutClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonZoomOutClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("축소 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 축소 실행을 취소 할 수 있다.

ButtonZoomOutClickAfter 이벤트

ButtonZoomOutClickAfter 이벤트는 축소 API가 호출 되어 축소 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

ButtonZoomOutClickAfter

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonZoomOutClickAfter">

alert("축소 API 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonZoomInClickBefore 이벤트

ButtonZoomInClickBefore 이벤트는 확대 API 가 호출되었을 때 확대 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

ButtonZoomInClickBefore(Cancel)

Cancel - True: 확대 명령을 실행함 (실행), False: 확대 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonZoomInClickBefore">
alert("확대 API 호출 이전 이벤트이다.");
</SCRIPT>
<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonZoomInClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

 $<\!\!\mathsf{SCRIPT}\ \mathsf{LANGUAGE} = \!\!\mathsf{"VBScript"}\ \mathsf{FOR} = \!\!\mathsf{"RQViewer"}\ \mathsf{EVENT} = \!\!\mathsf{"ButtonZoomInClickBefore}(\mathsf{Cancel}) \ \mathsf{">} \\$

Dim MyVar

MyVar = MsgBox("확대 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 확대 실행을 취소 할 수 있다.

ButtonZoomInClickAfter 이벤트



ButtonZoomInClickAfter 이벤트는 확대 API가 호출 되어 확대 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

ButtonZoomInClickAfter

확대 API 호출 이후 확대 명령 실행 이후 발생되는 이벤트

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonZoomInClickAfter">

alert("확대 API 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonMovePrevPageClickBefore 이벤트

ButtonMovePrevPageClickBefore 이벤트는 이전 페이지 이동 버튼이 클릭되었을 때 이전 페이지로 이동 명령을 실행하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

Button Move Prev Page Click Before (Cancel)

Cancel - True: 이전 페이지 이동 명령을 실행함 (실행), False: 이전 페이지 이동 명령을 취소함 (취소)

■ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonMovePrevPageClickBefore">

alert("이전 페이지 이동 실행 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonMovePrevPageClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonMovePrevPageClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("이전페이지로 이동 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 이전 페이지 이동 실행을 취소 할 수 있다.

ButtonMovePrevPageClickAfter 이벤트

ButtonMovePrevPageClickAfter 이벤트는 이전 페이지 이동 버튼이 클릭되어 이전 페이지 이동 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

Button Move PrevPage Click After

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonMovePrevPageClickAfter">

alert("이전 페이지 이동 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonMoveNextPageClickBefore 이벤트

ButtonMoveNextPageClickBefore 이벤트는 다음 페이지 이동 버튼이 클릭되었을 때 다음 페이지로 이동 명령을 실행하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

Button Move Next Page Click Before (Cancel)

Cancel - True: 다음 페이지 이동 명령을 실행함 (실행), False: 다음 페이지 이동 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용 예제

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonMoveNextPageClickBefore">

alert("다음 페이지 이동 실행 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonMovePrevPageClickBefore 이벤트만 발생한다.

▶ 사용 예제 - VBScript 사용

<HTML>

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonMoveNextPageClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("다음 페이지로 이동 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 다음 페이지 이동 실행을 취소 할 수 있다.

ButtonMoveNextPageClickAfter 이벤트

ButtonMoveNextPageClickAfter 이벤트는 다음 페이지 이동 버튼이 클릭되어 다음 페이지 이동 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

Button Move Next Page Click After

사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT=" ButtonMoveNextPageClickAfter">

alert("다음 페이지 이동 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonMoveFirstPageClickBefore 이벤트

ButtonMoveFirstPageClickBefore 이벤트는 처음 페이지 이동 버튼이 클릭되었을 때 처음 페이지로 이동 명령을 실행하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

ButtonMoveFirstPageClickBefore (Cancel)

Cancel - True: 처음 페이지 이동 명령을 실행함 (실행), False: 처음 페이지 이동 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonMoveNextPageClickBefore">

alert("처음 페이지 이동 실행 이전 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonMoveFirstPageClickBefore 이벤트만 발생한다.

사용 예제 - VBScript 사용

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonMoveFirstPageClickBefore(Cancel)">

Dim MyVar



MyVar = MsgBox("처음 페이지로 이동 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 처음 페이지 이동 실행을 취소 할 수 있다.

ButtonMoveFirstPageClickAfter 이벤트

ButtonMoveFirstPageClickAfter 이벤트는 처음 페이지 이동 버튼이 클릭되어 처음 페이지 이동 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

Button Move First Page Click After

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonMoveFirstPageClickAfter">

alert("처음 페이지 이동 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

ButtonMoveLastPageClickBefore 이벤트

ButtonMoveLastPageClickBefore 이벤트는 마지막 페이지 이동 버튼이 클릭되었을 때 마지막페이지로 이동 명령을 실행 하기 전에 발행하는 이벤트이다.

□ 이벤트 정의

 $Button Move Last Page Click Before\ (Cancel)$

Cancel - True: 마지막 페이지 이동 명령을 실행함 (실행), False: 마지막 페이지 이동 명령을 취소함 (취소)

▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonMoveLastPageClickBefore">
 alert("마지막 페이지 이동 실행 이전 이벤트이다.");
</SCRIPT>
<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

JavaScript 의 경우 ButtonMoveLastPageClickBefore 이벤트만 발생한다.

□ 사용 예제 - VBScript 사용 예제

<SCRIPT LANGUAGE="VBScript" FOR="RQViewer" EVENT=" ButtonMoveLastPageClickBefore(Cancel)">

Dim MyVar

MyVar = MsgBox("마지막 페이지로 이동 하시겠습니까?",vbOKCancel)

if (MyVar = vbCancel) Then

Cancel = True

End iF

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

VBScript 의 경우 Cancel 여부 인자를 넘겨 마지막 페이지 이동 실행을 취소 할 수 있다.

ButtonMoveLastPageClickAfter 이벤트

ButtonMoveLastPageClickAfter 이벤트는 마지막 페이지 이동 버튼이 클릭되어 마지막 페이지 이동 명령이 실행된 이후 발생하는 이벤트이다.

□ 이벤트 정의

Button Move Last Page Click After



▶ 사용 예제 - JavaScript 사용

<SCRIPT LANGUAGE="JavaScript" FOR="RQViewer" EVENT="ButtonMoveLastPageClickAfter"> alert("마지막 페이지 이동 호출 이후 이벤트이다.");

</SCRIPT>

<SCRIPT language=JavaScript SRC="./setup/rqviewer.js"></SCRIPT>

3.2.2. (RQReport)

SetRuntimeVariable

SetRuntimeVariable 함수는 리포트의 실행변수 값을 입력하는 함수이다.

▶ 함수 정의 및 인자

| SetRuntimeVariable([실행변수]) |
|----------------------------|
| 리포트 실행 시 사용할 실행변수의 값을 지정한다 |
| [실행변수] - 실행변수명 실행변수값 |

GetRuntimeVariable

GetRuntimeVariable 함수는 리포트의 실행변수를 가져오는 함수이다.

□ 함수 정의 및 인자

| [실행변수]=GetRuntimeVariable |
|---------------------------|
| 리포트에 설정된 실행변수를 가져온다. |

GetObject

GetObject 함수는 리포트의 설계 Object 를 가져오는 함수이다. – 추후 개발 예정



□ 함수 정의 및 인자

| [RQReport 객체] = GetObject([설계오브젝트명]) |
|--------------------------------------|
| 리포트에 사용된 설계 오브젝트를 가져오는 함수이다. |
| [RQReport 객체] = 문서에서 사용된 설계 오브젝트 객체 |
| [설계오브젝트명]= 문서 설계 시 지정된 설계오브젝트 이름 |

${\bf Get SQLC ontrol}$

GetSQLControl 함수는 리포트에서 사용된 SQL 관리 Control 객체를 얻어오는 함수이다. 이 RQSQLCtrl에는 문서 오 픈시 읽어온 SQL이 저장되어 있다.

▶ 함수 정의 및 인자

| [RQSQLControl 객체] = GetSQLControl() |
|---|
| 리포트에 사용된 SQL 관리 Control 객체를 가져오는 함수이다. |
| [RQSQLControl 객체] = 문서에서 사용된 SQL Control 객체 |

3.2.3 (RQObject)

SetProperty

SetProperty 함수는 리포트의 사용된 설계 오브젝트의 속성값을 변경하는 함수이다.[추후개발]

| SetProperty([속성명],[속성값]) |
|------------------------------------|
| 리포트에 사용된 설계 Object의 속성값을 변경하는 함수이다 |
| [속성명] - 설계 Object 의 속성 이름 |



GetProperty

GetProperty 함수는 리포트의 사용된 설계 오브젝트의 속성값을 가져오는 함수이다.[추후개발]

▶ 함수 정의 및 인자

| [속성값]=GetProperty([속성명]) |
|--------------------------------------|
| 리포트에 사용된 설계 Object 의 속성값을 가져오는 함수이다. |
| [속성값] = 설계 Object 의 속성 값 |
| [속성명] - 설계 Object 의 속성 이름 |

3.2.4 SQL (RQSQLControl)

GetCount

GetCount 함수는 리포트의 사용된 SQL(DataSet)의 개수를 가져오는 함수이다.

▶ 함수 정의 및 인자

| [SQL 갯수]= GetCount() |
|-------------------------------------|
| 리포트에 사용된 SQL 의 개수를 가져오는 함수이다 |
| [SQL 갯수] = 사용된 SQL의 갯수 |

$GetDefaultRowSeperator / \ GetDefaultColSeperator$

SQLControl 에 사용된 DataSet 의 Default Row/Col Seperator 를 반환한다.

| GetDefaultRowSeperator(), GetDefaultColSeperator() |
|--|
| DataSet 의 Default Seperator 값 |



| RowSep 기본값 = \n (줄바꿈) |
|-----------------------|
| ColSep 기본값 = \t (탭) |

GetDBString

GetDBString 함수는 리포트의 사용된 전체 DataBase 연결 정보를 XML 구조의 String 으로한번에 추출하는 함수이다.

▶ 함수 정의 및 인자

| [XML DB String] = GetDBString() |
|--|
| 리포트에 사용된 DB 연결 정보 전체를 XML String 형태로 리턴하는 함수 |
| [XML DB String] - 전체 DB 연결 정도 XML String |

GetSQLString

GetSQLString 함수는 리포트의 사용된 전체 SQL(Query) 정보를 XML 구조의 String 으로한번에 추출하는 함수이다.

□ 함수 정의 및 인자

| [XML SQL String] = GetSQLString() |
|--|
| 리포트에 사용된 SQL 정보 전체를 XML String 형태로 리턴하는 함수 |
| [XML SQL String] - 전체 SQL 정보 XML String |

GetQueryCount

GetQueryCount 함수는 리포트의 사용된 전체 SQL(Query) 개수를 리턴하는 함수이다.

| [Query Count] = GetQueryCount() |
|------------------------------------|
| 리포트에 사용된 SQL 전체 개수를 리턴하는 함수 |
| [Query Count] - 전체 Query Count |

GetDBIFType

GetDBIFType 함수는 리포트의 사용된 해당 DataSet 인덱스의 접속 DB 종류를 가져오는 함수이다.

▶ 함수 정의 및 인자

| [DB IF type] = GetDBIFType ([인덱스]) |
|---|
| 리포트에 사용된 해당 인덱스의 DB 종류를 가져오는 함수 |
| [DB IF Type] = 해당 인덱스의 GetDBIFType (1 = JDBC, 3 = SAP, 6 = XML) |
| [인덱스] - DataSet 인덱스 |

GetXMLRoot

GetXMLRoot 함수는 해당 인덱스의 XML DataSet File 의 Root 설정를 리턴하는 함수이다. (XML 연결시 사용)

▷ 함수 정의 및 인자

| [XML Root] = GetXMLRoot([Query Index]) |
|---|
| 해당 인덱스의 XML DataSet File 의 Root 를 리턴하는 함수 |
| [XML Root] - XML DataSet 의 XML Root |
| [Query Index] – XML DataSet Index |

GetConnectionString

GetConnectionString 함수는 리포트의 사용된 해당 DataSet 인덱스의 접속정보 ConnectionString 을 가져오는 함수이다.

| [ConnectionString] = GetConnectionString([인덱스]) |
|---|
| 리포트에 사용된 해당 인덱스의 ConnectionString 을 가져오는 함수 |
| [ConnectionString] = 해당 인덱스의 ConnectionString |
| [인덱스] - DataSet 인덱스 |



GetDBInfo

GetDBInfo 함수는 리포트의 사용된 해당 DataSet 인덱스의 DB 접속 정보 XML을 가져오는 함수이다.

▶ 함수 정의 및 인자

| [DBINFO XML String] = GetDBInfo ([인덱스]) |
|--|
| 리포트에 사용된 해당 인덱스의 DBINFO XML String 을 가져오는 함수 |
| [DBINFO XML String] = 해당 인덱스의 DBINFO XML String : <dbinfo> </dbinfo> |
| [인덱스] - DataSet 인덱스 |

GetSQLStmt

GetSQLStmt 함수는 리포트의 사용된 해당 DataSet 인덱스의 SQL을 XML 형태로 가져오는 함수이다.

▶ 함수 정의 및 인자

| [SQL XML String] = GetSQLStmt ([인덱스]) |
|--|
| 리포트에 사용된 해당 인덱스의 SQL XML String 을 가져오는 함수 |
| [SQL XML String] = 해당 인덱스의 SQL XML String : <sqlstmt> </sqlstmt> |
| [인덱스] - DataSet 인덱스 |

GetSQLStmtCount

GetSQLStmtCount 함수는 리포트의 사용된 해당 DataSet 인덱스의 SQL 내의 SQL 개수를 리턴해준다..

| [SQL 개수] = GetSQLStmtCount ([인덱스]) |
|---|
| 리포트에 사용된 해당 인덱스의 SQL 개수를 가져오는 함수 |
| [SQL 개수] = 해당 인덱스의 SQL 개수 |
| [인덱스] - DataSet 인덱스 |



GetDBID

GetDBID 함수는 리포트의 사용된 해당 DataSet 인덱스의 Database ID 를 리턴해준다..

▶ 함수 정의 및 인자

| [Database ID] = GetDBID ([인덱스]) |
|--|
| 리포트에 사용된 해당 인덱스의 Database ID 를 가져오는 함수 |
| [Database ID] = 해당 인덱스의 Database ID String |
| [인덱스] - DataSet 인덱스 |

GetRowSeperator

GetRowSeperator 함수는 리포트의 사용된 해당 DataSet 인덱스의 Row Seperator 를 리턴해준다. (TXT 연결시 사용)

□ 함수 정의 및 인자

| [Row Seperator] = GetRowSeperator ([인덱스]) |
|---|
| 리포트에 사용된 해당 인덱스의 Row Seperator 를 가져오는 함수 |
| [Row Seperator] = 해당 인덱스의 Row Seperator |
| [인덱스] - DataSet 인덱스 |

GetColSeperator

GetColSeperator 함수는 리포트의 사용된 해당 DataSet 인덱스의 Col Seperator 를 리턴해준다. (TXT 연결시 사용)

| [Col Seperator] = GetColSeperator ([인덱스]) |
|---|
| 리포트에 사용된 해당 인덱스의 Col Seperator를 가져오는 함수 |
| [Col Seperator] = 해당 인덱스의 Col Seperator |
| [인덱스] - DataSet 인덱스 |

GetEncoding

GetEncoding 함수는 리포트의 사용된 해당 DataSet 인덱스의 Encoding 을 리턴해준다. (TXT 연결시 사용)

▶ 함수 정의 및 인자

| [Encoding] = GetEncoding ([인덱스]) |
|--------------------------------------|
| 리포트에 사용된 해당 인덱스의 Encoding을 가져오는 함수 |
| [Encoding] = 해당 인덱스의 Encoding String |
| [인덱스] - DataSet 인덱스 |

3.2.5 RQConnectionCtrl

Interface (rqviewer_multi.jsp)

```
<@@ page contentType="text/html;charset=EUC-KR"%>
<%
         String path = request.getContextPath();
         String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
         String Im_doc = request.getParameter("doc");
         String doc = new String(Im_doc.getBytes("8859_1"),"KSC5601");
         String Im_runvar = request.getParameter("runvar");
         String runvar;
         if (Im_runvar != null)
                   runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
         else
                  runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
```



```
function OnRun(docName,runvar)
       var oReport;
       var oSQL;
       var oConnection;
       var oConnObj;
       var oDataSet;
       var oDataSetCtrl;
       //---Viewer 속성(Property) 세팅 부분------
       //RQViewer.BackColor = 255;
       //RQViewer.ToolBarVisible = false;
       //RQViewer.StatusBarVisible = false;
       //RQViewer.ShowProgressDialog = false;
       //RQViewer.EmptyDataCheckOption = "part"; //"all" or "part" or "Query1|Query2"
       //---OpenReport 를 사용하여 문서열기------
       if (docName == 'null')
       {
               alert("DocName is Null");
               return;
       }
       else
               try
               {
                   oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
               } catch(err)
               {
                       return;
               }
       }
       //---문서연결을 위한 BaseURL 설정------
```

```
RQViewer.SetBaseURL("<%=basePath%>");
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//---CreateConnectionControl 을 사용하여 Connection 개체 만들기------
oConnection = RQViewer.CreateConnectionControl();
//---Connection 개체의 설정-----
//[Type 1] : HTTP(REQUBE 서버 결과)와 XML(Web Server 에 있는 XML 파일) 혼합 연결
var nXMLCount = 0;
for (i=0; i<oSQL.GetCount(); i++)
        nType = oSQL.GetDBIFType(i);
       switch(nType)
        {
               case 6://XML
                       oConnObj = oConnection.GetConnectionByID("http.post",oSQL.GetDBID(i));
                       oConnObj.SetXMLRoot(nXMLCount,oSQL.GetXMLRoot(i));\\
                       oConnObj.AddResponseCount(1);
                       oConnObj.SetPath("<%=basePath%>xmlData.xml");
                       nXMLCount++;
               break;
               default: // HTTP(REQUBE Server 이용)
                       oConnObj = oConnection.GetConnectionByID("http.post",oSQL.GetDBID(i));
                       oConnObj.AddParameter("action", "getRsDB");
                       oConnObj.AddParameter("runvar",oReport.GetRuntimeVariable());
                       oConnObj.AddParameter("doc",docName);
                       oConnObj.AddResponseCount(oSQL. GetSQLStmtCount(i));
                       oConnObj.SetPath("<%=basePath%>RQDataset.jsp");
               break;
```

```
REQUBE REPORT Server 6.0
```

```
}
}
//[Type 2] : LOCAL(REQUBE 서버를 통하지 않는 XML Local File or String)과 TXT File or String
for (i=0; i<oSQL.GetCount(); i++)
{
        nType = oSQL.GetDBIFType(i);
        switch(nType)
        {
                 case 5: // TXT
                          oConnObj = oConnection.GetConnectionByID("local",oSQL.GetDBID(i));
                          oConnObj.SetEncoding(oSQL.GetEncoding(i));
                          oConnObj. SetSeperator (oSQL. GetRowSeperator (i), oSQL. GetColSeperator (i));\\
                          oConnObj.AddResponseFile(oSQL.GetConnectionString(i));
                          //oConnObj.AddResponseString(TXT_STRING_DATA);
                 break;
                 case 6: // XML
                          oConnObj = oConnection.GetConnectionByID("local",oSQL.GetDBID(i));
oConnObj. AddXMLResponseFile (oSQL.GetConnectionString (i), oSQL.GetXMLRoot (i));\\
                         //oConnObj.AddXMLResponseString(XML_STRING_DATA,oSQL.GetXMLRoot(i));
                 break;
        }
}
//---DataSet 을 설정-----
//--XML 파일과 TXT 파일 DataSet 이 모두 존재하는 경우
oDataSetCtrl = RQViewer.CreateDataSetControl();
for (i = 0; i< oConnection.GetCount();i++)
{
        oConnObj = oConnection.GetConnection(i);
        nType = oConnObj.GetResponseType();
        switch(nType)
```

```
REQUBE REPORT Server 6.0
```

```
{
                        case 2: // XML
                                 for (j = 0; j < oConnObj.GetCount(); j++)
                                         oDataSet =
oDataSetCtrl.SetDataSet(oConnObj.GetResponseIndex(j), "xml");\\
        oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j),oSQL.GetXMLRoot(j));
                                 }
                        break;
                        default: // TXT
                                 for (j = 0; j < oConnObj.GetCount(); j++)
                                         oDataSet =
oDataSetCtrl.SetDataSet(oConnObj.GetResponseIndex(j),"txt");
                                         oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j));
                                         oDataSet.SetEncoding(oConnObj.GetEncoding());
                                 oDataSet.SetSeperator(oConnObj.GetRowSeperator(),oConnObj.GetColSeperator());
                                         oDataSet.SetDataSetType(nType);
                                 }
                        break;
                }
        }
        //---문서실행-----
        RQViewer.Run();
//[----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
function RQ_HyperLinkSummit()
{
        RQ_HyperLink.submit();
```

```
function RQ_SetTarget(val)
{
        RQ_HyperLink.target = val;
function RQ_SetAction(val)
{
        RQ_HyperLink.action = val;
}
function RQ_AddValue(name,val)
{
        if(document.getElementById(name)!= null)
                 document.getElementById(name).value = val;
        }
        else{
                 elem = document.createElement("input");
                 elem.type="hidden";
                 elem.name=name;
                 elem.id=name;
                 elem.value=val;
                 RQ_HyperLink.appendChild(elem);
        }
}
function RQ_ExcuteHyperLink(url,target,option)
{
        window.open(url,target,option);
//]----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
</SCRIPT>
<!-- 이벤트 설정 -->
```

```
<script language="JavaScript" for="RQViewer" event="EmptyResultData">
        alert("EmptyResultData");
</script>
<script language="JavaScript" for="RQViewer" event="EndRunReport">
        //alert("EndRunReport");
</script>
<script language="JavaScript" for="RQViewer" event="Refresh">
        alert("Refresh");
</script>
<STYLE>
         DIV { padding-top: 0px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px;}
</STYLE>
</HEAD>
<BODY topmargin="0" leftmargin="0" onload="OnRun('<%=doc%>','<%=runvar%>')">
<script language="javaScript" SRC="./setup/rqviewer.js"></script>
<!--[ Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
<DIV name=RQ_HyperLinkPost height=0 width=0>
<FORM id=RQ_HyperLink method=post name=RQ_HyperLink>
</FORM>
</DIV>
<!--] Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
</BODY>
</HTML>
```

AddConnection

- AddConnection 함수는 리포트에서 사용할 Connection 들을 관리할 ConnectionCtrl 에 Connection 을 추가하고 Connection을 반환하는 함수이다.(단일 DB IF 사용시 주로 사용한다.)
- 생성한 추가대로 Connection 을 실행하여 결과 DataSet 을 가지고 온다.

□ 함수 정의 및 인자

[Connection 객체] = AddConnection([connection type])
Connection Type 에 따라 Connection 객체를 생성해 추가하고 반환한다.
[Connection 객체] = 생성된 Connection 객체
http.post - RQConnectionHTTP 객체
jdbc - RQConnectionJDBC 객체
local - RQConnectionLOCAL 객체
[connection type] – http.post : HTTP – POST
jdbc : JDBC Connection 을 이용
local : Client PC 내의 DataSet 파일.

GetConnection

GetConnection 함수는 RQConnectionCtrl 에서 AddConnection 을 통하여 추가된 순서에 따라 Connection 을 얻어 오는 함수이다. 해당 Connection을 가져와 설정을 변경하거나, 결과 DataSet 파일명 등을 얻어오기 위해 사용할 수 있다.

▶ 함수 정의 및 인자

[Connection 객체] = GetConnection([인덱스])
RQConnectionCtrl 에 추가된 순서에 따라 인덱스로 Connection 을 얻어온다.
[Connection 객체] = 해당 인덱스 위치의 Connection 객체
[인덱스] - 해당 Connection 인덱스

GetConnectionByName

- GetConnectionByName 함수는 리포트에서 사용할 Connection 들을 관리할 ConnectionCtrl 에서 해당 Connection 이 없을 경우 추가하고 있을경우 존재하는 Connection을 반환하는 함수이다.(다중 DB IF 사용시 사용, IF 의 종류가 모두 다를경우 주로 사용한다.)
- Connection 종류 별로 하나의 Connection 만 유지할 경우 사용된다. (여러 개의 동일한 종류의 Connection을 생성하는 것보다 실행 속도 향상에 도움이 된다.)



[Connection 객체] = GetConnectionByName([connection type])
Connection Type 에 따라 Connection 객체를 없을경우 생성해 추가하고 반환하고, 존재 할 경우 존재하는
Connection 객체를 리턴한다.
[Connection 객체] = 생성된 Connection 객체
http.post - RQConnectionHTTP 객체
jdbc - RQConnectionJDBC 객체
local - RQConnectionLOCAL 객체
[connection type] – http.post : HTTP – POST
jdbc : JDBC Connection 을 이용
local : Client PC 내의 DataSet 파일.

GetConnectionByID

- GetConnectionByID 함수는 리포트에서 사용할 Connection 들을 관리할 ConnectionCtrl 에서 해당 ConnectionType 과 ConnectionID가 없을 경우 추가하고 있을경우 존재하는 Connection을 반환하는 함수이다. (다중 DB IF 사용시 사용, IF 의 종류가 유사한 경우가 있어, DBID로 구분이 필요할 경우 사용한다.)
- Connection 종류 및 ConnectionID 별로 하나의 Connection 만 유지할 경우 사용된다. (여러 개의 동일한 종류의 Connection을 생성하는 것보다 실행 속도 향상에 도움이 된다.)

▶ 함수 정의 및 인자

[Connection 객체] = GetConnectionByID([connection type],[connection id)
Connection Type 에 따라 Connection 객체를 없을경우 생성해 추가하고 반환하고, 존재 할 경우 존재하는
Connection 객체를 리턴한다.
[Connection 객체] = 생성된 Connection 객체
http.post - RQConnectionHTTP 객체
jdbc - RQConnectionJDBC 객체
local - RQConnectionLOCAL 객체
[connection type] – http.post : HTTP – POST
jdbc : JDBC Connection 을 이용
local : Client PC 내의 DataSet 파일.
[connection id] – connection string 과 같이 string 형태의 id:

GetConnectionType



GetConnectionType 함수는 ConnectionControl 내의 해당 인덱스에 해당하는 Connection의 종류를 반환한다.

□ 함수 정의 및 인자

[Connection 종류] = GetConnectionType ([Connection Index])
RQConnectionCtrl 에 추가된 Connection 의 종류를 반환한다.
[Connection 종류] - Connection 종류 (1 = http.post, 3 = JDBC, 4 = LOCAL, 5 = SAP)
[Connection Index]

Send

Send 함수는 AddConnection 을 통하여 추가된 순서에 따라 Connection 을 실행하여 결과 DataSet 파일을 받아오는 기능을 한다. RQViewer Control 의 Run 함수 사용시는 사용하지 않는다.

□ 함수 정의 및 인자

Send ()
RQConnectionCtrl 에 추가된 순서에 따라 Connection 을 실행하여 결과 DataSet 을 얻어온다.
없음

GetCount

GetCount 함수는 ConnectionControl 내의 Connection 의 갯수를 반환한다.

▶ 함수 정의 및 인자

[Connection 갯수] = GetCount()
RQConnectionCtrl 에 추가된 Connection 의 개수를 반환한다.
없음

3.2.6 RQConnectionHTTP

HTTP REQUBE Server (rqviewer_http.jsp)

```
<@ page contentType="text/html;charset=EUC-KR"%>
<%
         String path = request.getContextPath();
         String basePath = request.getScheme()+":"/"+request.getServerName()+":"+request.getServerPort()+path+"/";
         String Im_doc = request.getParameter("doc");
         String doc = new String(Im_doc.getBytes("8859_1"),"KSC5601");
         String Im_runvar = request.getParameter("runvar");
         String runvar;
         if (Im_runvar != null)
                  runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
         else
                  runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
function OnRun(docName,runvar)
         var oReport;
         var oSQL;
         var oConnection;
         var OConnObj;
         var oDataSet;
         var oDataSetCtrl;
         //---Viewer 속성(Property) 세팅 부분-----
         //RQViewer.BackColor = 255;
         //RQViewer.ToolBarVisible = false;
         //RQViewer.StatusBarVisible = false;
```

```
//RQViewer.ShowProgressDialog = false;
//RQViewer.EmptyDataCheckOption = "part"; //"all" or "part" or "Query1|Query2"
//---OpenReport 를 사용하여 문서열기------
if (docName == 'null')
{
       alert("DocName is Null");
       return;
}
else
      try
       {
          oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
      } catch(err)
       {
              return;
      }
}
//---문서연결을 위한 BaseURL 설정------
RQViewer.SetBaseURL("<%=basePath%>");
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//---CreateConnectionControl 을 사용하여 Connection 개체 만들기------
oConnection = RQViewer.CreateConnectionControl();
//---Connection 개체의 설정------
OConnObj = oConnection.AddConnection("http.post");
OConnObj.AddParameter("action", "getRsDB");
```

```
OConnObj.AddParameter("runvar",oReport.GetRuntimeVariable());
       OConnObj.AddParameter("doc",docName);
       OConnObj.SetQueryCount(oSQL.GetQueryCount());
       OConnObj.SetPath("<%=basePath%>RQDataset.jsp");
       //---DataSet 을 설정-----
       oDataSetCtrl = RQViewer.CreateDataSetControl();
       for (i = 0; i< oConnection.GetCount();i++)
               oConnObj = oConnection.GetConnection(i);
               for ( j = 0; j < oConnObj.GetCount(); j++)
                       oDataSet = oDataSetCtrl.AddDataSet("txt");
                       oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j));
                       oDataSet.SetEncoding(oConnObj.GetEncoding());
                       oDataSet.SetSeperator (oConnObj.GetRowSeperator (), oConnObj.GetColSeperator ());\\
               }
       }
       //---문서실행-----
       RQViewer.Run();
//[----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
function RQ_HyperLinkSummit()
{
       RQ_HyperLink.submit();
function RQ_SetTarget(val)
        RQ_HyperLink.target = val;
```

```
function RQ_SetAction(val)
{
        RQ_HyperLink.action = val;
}
function RQ_AddValue(name,val)
{
        if(document.getElementById(name)!= null)
        {
                 document.getElementById(name).value = val;
        }
        else{
                 elem = document.createElement("input");
                 elem.type="hidden";
                 elem.name=name;
                 elem.id=name;
                 elem.value=val;
                 RQ_HyperLink.appendChild(elem);
        }
}
function RQ_ExcuteHyperLink(url,target,option)
{
        window.open(url,target,option);
//]----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
-->
</SCRIPT>
<!-- 이벤트 설정 -->
<script language="JavaScript" for="RQViewer" event="EmptyResultData">
        alert("EmptyResultData");
</script>
```

```
<script language="JavaScript" for="RQViewer" event="EndRunReport">
        alert("EndRunReport");
</script>
<script language="JavaScript" for="RQViewer" event="Refresh">
        alert("Refresh");
</script>
<STYLE>
        DIV { padding-top: 0px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px;}
</STYLE>
</HEAD>
<BODY topmargin="0" leftmargin="0" onload="OnRun('<%=doc%>','<%=runvar%>')">
<script language="javaScript" SRC="./setup/rqviewer.js"></script>
<!--[ Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
<DIV name=RQ_HyperLinkPost height=0 width=0>
<FORM id=RQ_HyperLink method=post name=RQ_HyperLink>
</FORM>
</DIV>
<!--] Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
</BODY>
</HTML>
```

HTTP XML (rqviewer_http_xml.jsp)

```
<%@ page contentType="text/html;charset=EUC-KR"%>

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

String lm_doc = request.getParameter("doc");

String doc = new String(lm_doc.getBytes("8859_1"),"KSC5601");

String lm_runvar = request.getParameter("runvar");

String runvar;

if (lm_runvar != null)
```

```
runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
        else
                runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
function OnRun(docName,runvar)
{
        var oReport;
        var oSQL;
        var oConnection;
        var OConnObj;
        var oDataSet;
        var oDataSetCtrl;
        //---Viewer 속성(Property) 세팅 부분-----
        //RQViewer.BackColor = 255;
        //RQViewer.ToolBarVisible = false;
        //RQViewer.StatusBarVisible = false;
        //RQViewer.ShowProgressDialog = false;
        //RQViewer.EmptyDataCheckOption = "part"; //"all" or "part" or "Query1|Query2"
        //---OpenReport 를 사용하여 문서열기------
        if (docName == 'null')
        {
                alert("DocName is Null");
                return;
        }
        else
```

```
try
       {
          oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
       } catch(err)
       {
              return;
       }
}
//---문서연결을 위한 BaseURL 설정------
RQViewer.SetBaseURL("<%=basePath%>");
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//--CreateConnectionControl 을 사용하여 Connection 개체 만들기------
oConnection = RQViewer.CreateConnectionControl();
//[XML] Web Server 의 XML Data 연결
oConnObj = oConnection.AddConnection("http.post");
for (i=0; i<oSQL.GetCount(); i++)
{
       oConnObj.AddResponseCount(1);
       oConnObj.SetXMLRoot(i,oSQL.GetXMLRoot(i));\\
}
oConnObj.SetPath("<%=basePath%>xmlData.xml");
//---DataSet 을 설정-----
//[XML] DataSet 설정
oDataSetCtrl = RQViewer.CreateDataSetControl();
```

```
for (i = 0; i< oConnection.GetCount();i++)
                oConnObj = oConnection.GetConnection(i);
               for ( j = 0; j < oConnObj.GetCount(); j++)
                        oDataSet = oDataSetCtrl.AddDataSet("xml");
                        oDataSet.SetDataSetFile (oConnObj.GetResponseFile (j), oConnObj.GetXMLRoot (j));\\
                                                               // XML Element 순서로 연결 옵션
                       //oDataSet.SetBindingOption(2);
               }
       }
       //---문서실행-----
        RQViewer.Run();
}
//[----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
function RQ_HyperLinkSummit()
{
        RQ_HyperLink.submit();
function RQ_SetTarget(val)
{
        RQ_HyperLink.target = val;
function RQ_SetAction(val)
{
        RQ_HyperLink.action = val;
function RQ_AddValue(name,val)
```

```
if(document.getElementById(name)!= null)
        {
                 document.getElementById(name).value = val;
        }
        else{
                  elem = document.createElement("input");
                  elem.type="hidden";
                  elem.name=name;
                 elem.id=name;
                 elem.value=val;
                 RQ_HyperLink.appendChild(elem);
        }
function RQ_ExcuteHyperLink(url,target,option)
{
        window.open(url,target,option);
//]----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
-->
</SCRIPT>
<!-- 이벤트 설정 -->
<script language="JavaScript" for="RQViewer" event="EmptyResultData">
        alert("EmptyResultData");
</script>
<script language="JavaScript" for="RQViewer" event="EndRunReport">
        alert("EndRunReport");
</script>
<script language="JavaScript" for="RQViewer" event="Refresh">
        alert("Refresh");
</script>
<STYLE>
```



```
DIV { padding-top: 0px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px;}
</STYLE>
</HEAD>
<BODY topmargin="0" leftmargin="0" onload="OnRun('<%=doc%>','<%=runvar%>')">
<script language="javaScript" SRC="./setup/rqviewer.js"></script>
<!-- [ Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
<DIV name=RQ_HyperLinkPost height=0 width=0>
<FORM id=RQ_HyperLink method=post name=RQ_HyperLink>
</FORM>
</DIV>
<!-- ] Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->
</BODY>
</HTML>
```

GetPath

GetPath 함수는 SetPath 를 이용하여 설정된 DataSet Server 의 URL 을 얻어온다.

▶ 함수 정의 및 인자

[Dataset Server URL] = GetPath()
SetPath 로 설정된 DataSet Server 의 URL 을 얻어온다.
없음

SetPath

SetPath 함수는 DataSet Server 의 URL 을 설정한다.

SetPath([Dataset Server URL])



DataSet Server 의 URL 을 설정한다.
없음

Response

Response 함수는 결과 요청에 대한 응답을 보기 위한 함수로, 주로 디버깅용으로 사용한다.

□ 함수 정의 및 인자

[Dataset Server URL] = Response()
DataSet Server 의 URL 을 설정한다
없음

AddParameter

AddParameter 함수는 서버에 보내는 요청에서 웹 호출 Parameter 를 추가하는 함수이다.

□ 함수 정의 및 인자

AddParameter([파라미터명],[파라미터값])
서버에 요청하는 호출의 파라미터를 추가하는 함수이다.
http:// /RQDataSet.jsp?[파라미터명]=[파라미터값]
[파라미터 명] - 호출 파라미터 명
[파라미터 값] - 호출 파라미터 값

GetParameter

GetParameter 함수는 서버에 보내는 요청에서 웹 호출 Parameter 를 추가하는 함수이다.

[파라미터값] = GetParameter([파라미터명])
[



서버에 요청하는 호출의 파라미터의 값을 가져오는 함수이다.
http:// /RQDataSet.jsp?[파라미터명]=[파라미터값]
[파라미터 값]= 호출 파라미터 값
[파라미터 명]= 호출 파라미터 명

Get Response File

GetResponseFile 함수는 내려 받은 DataSet 의 파일명을 가져오는 함수이다.

□ 함수 정의 및 인자

[DataSet 파일명] = GetResponseFile([인덱스])
서버에서 내려 받은 DataSet 파일명을 가져오는 함수이다.
[DataSet 파일명] = 내려받은 DataSet 파일명
[인덱스] - SQL 이 추가된 순서 인덱스

GetEncoding

GetEncoding 함수는 서버에서 내려 받은 결과 DataSet 의 Encoding 을 반환한다.

▶ 함수 정의 및 인자

[Encoding Type] = GetEncoding()
서버에서 받은 결과 DataSet 의 Encoidng 을 반환한다.
[Encoding Type] = Encoding 값
(utf-8, utf-16, ansi)

RemoveParameter

RemoveParameter 함수는 서버에 보내는 요청에서 웹 호출 Parameter 를 제거하는 함수이다.

RemoveParameter([파라미터명])
1 to 110 to 1 diameter ([=1=1=1=1])



서버에 요청하는 호출의 파라미터의 값을 삭제하는 함수이다.
[파라미터 명]- 호출 파라미터 명

GetCount

GetCount 함수는 서버에서 내려 받은 결과 DataSet 의 갯수를 반환한다.

▶ 함수 정의 및 인자

[DataSet 개수] = GetCount()
서버에서 받은 결과 DataSet 의 갯수를 반환한다.
[DataSet 개수] = 내려받은 DataSet 개수

ReplaceParameter

ReplaceParameter 함수는 서버에 보내는 요청에서 웹 호출 Parameter 를 변경하는 함수이다.

▷ 함수 정의 및 인자

ReplaceParameter ([파라미터명],[파라미터값])
서버에 요청하는 호출의 파라미터의 값을 삭제하는 함수이다.
[파라미터 명] - 호출 파라미터 명
[파라미터 값] - 호출 파라미터 값

SetQueryCount

SetQueryCount 함수는 Connection 에서 사옹되는 Query 의 개수를 세팅한다..

SetQueryCount ([Query Count])
Connection 에서 사용되는 Query 의 개수를 세팅한다



[Query Count] - Query 개수 . (Connection 에서 나오는 Response File 의 수 세팅)

SetEncoding

SetEncoding 함수는 서버에서 내려 받은 결과 DataSet 의 Encoding 을 반환한다.

▶ 함수 정의 및 인자

SetEncoding([Encoding Type])
서버에 요청하는 Encoidng 을 세팅한다.
[Encoding Type] - Encoding 값
(utf-8, utf-16, ansi)

GetURL

GetURL 함수는 Connection 에 설정된 URL 을 반환한다.

▶ 함수 정의 및 인자

[URL] = GetURL()
Connection 에 설정된 URL을 리턴한다
없음

${\bf AddResponseCount}$

AddResponseCount 함수는 Connection 에 Query Count(Response File 개수)를 추가한다.

AddResponseCount ([Response Count])
Connection 에 Query (Response File)의 개수를 추가한다.
[Response Count] = Response File 의 개수

GetResponseIndex

GetResponseIndex 함수는 Connection 내의 AddResponseCount 로 추가된 Query Index 를 전체 Response Index 로 변환하여 반환한다.

▶ 함수 정의 및 인자

[Response Index] = GetResponseIndex ([Query Index])
Connection 에 Add 된 Query Index 를 전체 Response Index 로 변환하여 반환한다.
(멀티 Connection 일 때 사용한다.)
[Response Index] – 전체 Connnection Control 에 Add 된 Response Index 를 리턴
[Query Index] – Connection 에 Add 된 Query Index

GetResponseType

GetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 을 반환한다. (다중 IF 연결시 사용)

▷ 함수 정의 및 인자

[Response Type] = GetResponseType ()
Connection 에서 리턴 되는 DataSet 의 DataType 을 반환한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

SetResponseType

SetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 의 종류를 설정한다. (다중 IF 연결시 사용)

SetResponseType ([Response Type])
Connection 에서 리턴 되는 DataSet 의 DataType 을 설정한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

SetXMLRoot

SetXMLRoot 함수는 Connection 의 해당 Index 에 XML Root 정보를 추가한다. (웹서버에서 XML Data 를 얻는경우 사용)

▶ 함수 정의 및 인자

SetXMLRoot ([Index],[XML Root])
Connection 의 해당 Index 에 XMLRoot 정보를 추가한다
[Index] = Connection 에서 추가된 Response 의 Index(순차적으로 세팅한다)
[XML Root] = XML Root String

GetXMLRoot

GetXMLRoot 함수는 Connection 의 해당 Index 에 XML Root 정보를 얻어온다. (웹서버에서 XML Data 를 얻는경우 사용)

□ 함수 정의 및 인자

[XML Root] = GetXMLRoot ([Index])
Connection 의 해당 Index 에 XMLRoot 정보를 얻어온다
[XML Root] = XML Root String
[Index] = Connection 에서 추가된 Response 의 Index(순차적으로 세팅한다)

SetSeperator

SetSeperator 함수는 Connection 의 Row Seperator 및 Col Seperator 를 추가한다. (웹서버에서 오는 TXT 데이터가 기본 구분자 ('\n', '\t') 를 사용하지 않을 경우 사용한다. Connection 당 한가지만 설정할 수 있다.)

SetSeperator ([Row Seperator],[Col Seperator])
Connection 의 Row, Col Seperator 정보를 추가한다



[Row Seperator] = Connection 의 Row Seperator
[Col Seperator] = Connection 의 Col Seperator

GetRowSeperator

GetRowSeperator 함수는 Connection 의 Row Seperator 를 리턴한다. (DataSet 의 Seperator 설정시 사용)

□ 함수 정의 및 인자

[Row Seperator] = GetRowSeperator ()
Connection 에 설정된 Row Seperator 를 리턴한다. 특별히 설정하지 않으면 기본값 ('\n')을 리턴한다.
[Row Seperator] – Connection 에 설정된 Row Seperator

${\bf Get Col Seperator}$

GetColSeperator 함수는 Connection 의 Col Seperator 를 리턴한다. (DataSet 의 Seperator 설정시 사용)

▷ 함수 정의 및 인자

[Col Seperator] = GetColSeperator ()
Connection 에 설정된 Col Seperator 를 리턴한다. 특별히 설정하지 않으면 기본값 ('\t')을 리턴한다.
[Col Seperator] – Connection 에 설정된 Col Seperator

${\bf 3.2.7}\ RQC onnection JDBC/RQC onnection SAP$

REQUBE Report Designer 가

가

Response

Response 함수는 결과 요청에 대한 응답을 보기 위한 함수로, 주로 디버깅용으로 사용한다.



□ 함수 정의 및 인자

[Response 결과] = Response()
결과 요청에 대한 응답을 반환한다.
없음

GetEncoding

GetEncoding 함수는 내려 받은 결과 DataSet 의 Encoding 을 반환한다.

□ 함수 정의 및 인자

[Encoding Type] = GetEncoding()
받은 결과 DataSet 의 Encoidng 을 반환한다
[Encoding Type] = Encoding 값
(utf-8, utf-16, ansi)

GetResponseFile

GetResponseFile 함수는 내려 받은 DataSet 의 파일명을 가져오는 함수이다.

▶ 함수 정의 및 인자

[DataSet 파일명] = GetResponseFile([인덱스])
내려 받은 DataSet 파일명을 가져오는 함수이다.
[DataSet 파일명] = 내려받은 DataSet 파일명
[인덱스] - SQL 이 추가된 순서 인덱스

GetCount

GetCount 함수는 서버에서 내려 받은 결과 DataSet 의 갯수를 반환한다.



▶ 함수 정의 및 인자

[DataSet 개수] = GetCount()
서버에서 받은 결과 DataSet 의 갯수를 반환한다.
[DataSet 개수] = 내려받은 DataSet 개수

SetDBString

SetDBString 함수는 리포트의 사용된 전체 DataBase 연결 정보를 XML 구조의 String 으로한번에 세팅하는 함수이다.

□ 함수 정의 및 인자

SetDBString(XML DB String)
리포트에 사용된 DB 연결 정보 전체를 XML String 형태로 세팅하는 함수
(XML DB String) - 전체 DB 연결 정도 XML String

SetQueryString

SetQueryString 함수는 리포트의 사용된 전체 SQL(Query) 정보를 XML 구조의 Strin 을 한번에 세팅하는 함수이다.

▶ 함수 정의 및 인자

SetQueryString(XML SQL String)
리포트에 사용된 SQL 정보 전체를 XML String 형태로 세팅하는 함수
(XML SQL String) - 전체 SQL 정보 XML String

SetQueryCount

SetQueryCount 함수는 리포트의 사용된 전체 SQL(Query) 개수를 세팅하는 함수이다.(SetQueryString 과 사용한다.)

SetQueryCount(Query Count)
SetQuery Count()



리포트에 사용된 SQL 전체 개수를 세팅하는 함수
(Query Count) - 전체 Query Count

SetDBInfo

SetDBInfo 함수는 Connection 에 XML String 형태의 DB 정보를 세팅한다.

▶ 함수 정의 및 인자

SetDBInfo([DBINFO XML String])
Connection 에 DBINFO XML String 을 설정한다.
[DBINFO XML String] – DB INFO String <dbinfo></dbinfo>

SetSQLStmt

SetSQLStmt 함수는 Connection 에 XML String 형태의 SQL 정보를 세팅한다.

▶ 함수 정의 및 인자

SetSQLStmt ([SQL XML String])
Connection 에 SQL XML String 을 설정한다.
[SQL XML String] – DB INFO String <sqlstmt></sqlstmt>

GetResponseIndex

GetResponseIndex 함수는 Connection 내의 AddResponseCount 로 추가된 Query Index 를 전체 Response Index 로 변환하여 반환한다.

▶ 함수 정의 및 인자

[Response Index] = GetResponseIndex ([Query Index])



Connection 에 Add 된 Query Index 를 전체 Response Index 로 변환하여 반환한다.
(멀티 Connection 일 때 사용한다.)
[Response Index] – 전체 Connnection Control 에 Add 된 Response Index 를 리턴
[Query Index] – Connection 에 Add 된 Query Index

GetResponseType

GetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 을 반환한다. (다중 IF 연결시 사용)

▶ 함수 정의 및 인자

[Response Type] = GetResponseType ()
Connection 에서 리턴 되는 DataSet 의 DataType 을 반환한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

Set Response Type

SetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 의 종류를 설정한다. (다중 IF 연결시 사용)

□ 함수 정의 및 인자

SetResponseType ([Response Type])
Connection 에서 리턴 되는 DataSet 의 DataType 을 설정한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

AddDBInfo

AddDBInfo 함수는 Connection 에 XML String 형태의 DB 정보를 추가한다.

□ 함수 정의 및 인자

AddDBInfo([DBINFO XML String])
Connection 에 DBINFO XML String 을 추가한다.



[DBINFO XML String] - DB INFO String <DBInfo>...</DBInfo>

AddSQLStmt

AddSQLStmt 함수는 Connection 에 XML String 형태의 SQL 정보를 세팅한다.

▶ 함수 정의 및 인자

AddSQLStmt ([SQL XML String])
Connection 에 SQL XML String 을 설정한다.
[SQL XML String] – SQL Stmt String <sqlstmt></sqlstmt>

AddResponseCount

AddResponseCount 함수는 Connection 에 Response File 의 개수를 추가한다. (AddSQLStmt 과 함께 사용한다.)

□ 함수 정의 및 인자

AddResponseCount ([SQL XML String])
Connection 에 SQL XML String 을 설정한다.
[SQL XML String] – SQL Stmt String <sqlstmt></sqlstmt>

3.2.8 RQConnectionLOCAL

LOCAL (rqviewer_local.jsp)

 $<\!\!\%@\ page\ content{Type="text/html}; charset=EUC-KR"\%\!\!>$

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

String Im_doc = request.getParameter("doc");

```
String doc = new String(Im_doc.getBytes("8859_1"),"KSC5601");
        String Im_runvar = request.getParameter("runvar");
        String runvar;
        if (Im_runvar != null)
                 runvar = new String(Im_runvar.getBytes("8859_1"),"KSC5601");
        else
                 runvar = new String();
%>
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=EUC-KR">
<TITLE>REQUBE REPORT</TITLE>
<SCRIPT language="javaScript">
<!--
function OnRun(docName,runvar)
{
        var oReport;
        var oSQL;
        var oConnection;
        var OConnObj;
        var oDataSet;
        var oDataSetCtrl;
        //---Viewer 속성(Property) 세팅 부분-----
        //RQViewer.BackColor = 255;
        //RQViewer.ToolBarVisible = false;
        //RQViewer.StatusBarVisible = false;
        //RQViewer.ShowProgressDialog = false;
        //RQViewer.EmptyDataCheckOption = "part"; //"all" or "part" or "Query1|Query2"
        //---OpenReport 를 사용하여 문서열기------
        if (docName == 'null')
        {
                 alert("DocName is Null");
```

```
return;
}
else
{
       try
       {
          oReport = RQViewer.OpenReport("<%=basePath%>document/getreport.jsp?doc=" + docName);
       } catch(err)
       {
              return;
       }
}
//---문서연결을 위한 BaseURL 설정------
RQViewer.SetBaseURL("<%=basePath%>");
//---실행변수 설정-----
oReport.SetRuntimeVariable(runvar);
oSQL = oReport.GetSQLControl();
//---CreateConnectionControl 을 사용하여 Connection 개체 만들기------
oConnection = RQViewer.CreateConnectionControl();
//[Type 1] TXT
//---Connection 개체의 설정-----
//[TXT] File or String 연결
for (i=0; i<oSQL.GetCount(); i++)
{
       oConnObj = oConnection.AddConnection("local");
       oConnObj.SetEncoding(oSQL.GetEncoding(i));
       oConnObj.SetSeperator(oSQL.GetRowSeperator(i),oSQL.GetColSeperator(i));
       oConnObj.AddResponseFile(oSQL.GetConnectionString(i));
       //oConnObj.AddResponseString(TXT_STRING_DATA);
```

```
REQUBE REPORT Server 6.0
```

```
}
//---DataSet 을 설정------
//[TXT] DataSet 설정
oDataSetCtrl = RQViewer.CreateDataSetControl();
for (i = 0; i< oConnection.GetCount();i++)
{
        oConnObj = oConnection.GetConnection(i);
        for (j = 0; j < oConnObj.GetCount(); j++)
                oDataSet = oDataSetCtrl.AddDataSet("txt");
                oDataSet.SetDataSetFile(oConnObj.GetResponseFile(j));
                oDataSet.SetEncoding(oConnObj.GetEncoding());
                oDataSet.SetSeperator (oConnObj.GetRowSeperator (), oConnObj.GetColSeperator ());\\
                oDataSet.SetDataSetType(oConnObj.GetResponseType());
        }
}
//[Type 2] XML
//[XML] File or String 연결
for (i=0; i<oSQL.GetCount(); i++)
{
        oConnObj = oConnection.AddConnection("local");
        oConnObj.AddXMLResponseFile(oSQL.GetConnectionString(i),oSQL.GetXMLRoot(i));
        //oConnObj.AddXMLResponseString(XML_STRING_DATA,oSQL.GetXMLRoot(i));
}
//---DataSet 을 설정-----
//[XML] DataSet 설정
oDataSetCtrl = RQViewer.CreateDataSetControl();
```

```
for (i = 0; i< oConnection.GetCount();i++)
        {
                oConnObj = oConnection.GetConnection(i);
                for ( j = 0; j < oConnObj.GetCount(); j++)
                {
                        oDataSet = oDataSetCtrl.AddDataSet("xml");
                        oDataSet.SetDataSetFile (oConnObj.GetResponseFile (j), oConnObj.GetXMLRoot (j));\\
                        //oDataSet.SetBindingOption(2);
                                                                 // XML Element 순서로 연결 옵션
                }
        }
        //---문서실행-----
        RQViewer.Run();
//[----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분------
function RQ_HyperLinkSummit()
{
        RQ_HyperLink.submit();
function RQ_SetTarget(val)
{
        RQ_HyperLink.target = val;
}
function RQ_SetAction(val)
{
        RQ_HyperLink.action = val;
function RQ_AddValue(name,val)
{
        if(document.getElementById(name)!= null)
```

```
{
                  document.getElementById(name).value = val;
        }
        else{
                  elem = document.createElement("input");
                  elem.type="hidden";
                  elem.name=name;
                  elem.id=name;
                  elem.value=val;
                  RQ_HyperLink.appendChild(elem);
        }
}
function RQ_ExcuteHyperLink(url,target,option)
{
        window.open(url,target,option);
//]----Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분-----
-->
</SCRIPT>
<!-- 이벤트 설정 -->
<script language="JavaScript" for="RQViewer" event="EmptyResultData">
        alert("EmptyResultData");
</script>
<script language="JavaScript" for="RQViewer" event="EndRunReport">
        alert("EndRunReport");
</script>
<script language="JavaScript" for="RQViewer" event="Refresh">
        alert("Refresh");
</script>
<STYLE>
         DIV { padding-top: 0px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px;}
```



</STYLE>

</HEAD>

<BODY topmargin="0" leftmargin="0" onload="OnRun('<%=doc%>','<%=runvar%>')">

<script language="javaScript" SRC="./setup/rqviewer.js"></script>

<!--[Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->

<DIV name=RQ_HyperLinkPost height=0 width=0>

<FORM id=RQ_HyperLink method=post name=RQ_HyperLink>

</FORM>

</DIV>

<!--] Hyper Link(문서연결)을 위해 반드시 있어야 하는 부분 -->

</BODY>

</HTML>

GetEncoding

GetEncoding 함수는 결과 DataSet 의 Encoding 을 반환한다.

▶ 함수 정의 및 인자

[Encoding Type] = GetEncoding()
받은 결과 DataSet 의 Encoidng 을 반환한다.
[Encoding Type] = Encoding 값
(utf-8, utf-16, ansi)

GetResponseFile

GetResponseFile 함수는 DataSet 의 파일명을 가져오는 함수이다.

▷ 함수 정의 및 인자

[DataSet 파일명] = GetResponseFile([인덱스])



DataSet 파일명을 가져오는 함수이다.
[DataSet 파일명] = 내려받은 DataSet 파일명
[인덱스] - 파일이 추가된 순서 인덱스

GetCount

GetCount 함수는 추가된 결과 DataSet 의 갯수를 반환한다.

▶ 함수 정의 및 인자

[DataSet 개수] = GetCount()
추가된 결과 DataSet 의 갯수를 반환한다.
[DataSet 개수] = 내려받은 DataSet 개수

AddResponseFile

AddResponseFile 함수는 DataSet 의 파일명을 순서대로 추가하는 함수이다.

▶ 함수 정의 및 인자

AddResponseFile([DataSet 파일명])
DataSet 파일명을 추가하는 함수이다.
[DataSet 파일명] - 추가하는 DataSet 파일명

SetEncoding

SetEncoding 함수는 결과 DataSet 의 Encoding 을 설정한다. 하나의 Connection 에는 Encoding 을 하나만 설정할 수 있다. 따라서 여러 Encoding 을 포함하기 위해서는 여러 개의 Connection을 만들어야 한다.

▶ 함수 정의 및 인자

SetEncoding([Encoding Type])
결과 DataSet 의 Encoidng 을 설정한다.



[Encoding Type] - Encoding 값
(utf-8, utf-16, ansi)

AddXMLResponseFile

AddXMLResponseFile 함수는 XML DataSet 의 파일명과 XML Root 를 추가하는 함수이다.

▶ 함수 정의 및 인자

AddXMLResponseFile([DataSet 파일명],[XML Root])
XML DataSet 파일명 과 XML Root 를 추가하는 함수이다
[DataSet 파일명] - 추가하는 XML DataSet 파일명
[XML Root] - 추가하는 XML 파일의 반복 Root

AddXMLResponseString

AddXMLResponseString 함수는 XML DataSet 의 스트링과 XML Root 를 추가하는 함수이다.

□ 함수 정의 및 인자

AddXMLResponseString ([XML DataSet 스트링],[XML Root])
XML DataSet 스트링 과 XML Root 를 추가하는 함수이다
[XML DataSet 스트링] - 추가하는 XML DataSet 스트링
[XML Root] - 추가하는 XML 파일의 반복 Root

GetXMLRoot

GetXMLRoot 함수는 Connection 객체의 해당 Index 의 XML Root 를 리턴한다..

□ 함수 정의 및 인자

[XML Root] = GetXMLRoot ([인덱스])
Connection 객체에서 해당 Index 의 XML Root 를 리턴한다



[XML Root] = 해당 Index 의 XML Root
[인덱스] - 파일이 추가된 순서 인덱스

GetResponseIndex

GetResponseIndex 함수는 Connection 내의 AddResponseCount 로 추가된 Query Index 를 전체 Response Index 로 변환하여 반환한다.

□ 함수 정의 및 인자

[Response Index] = GetResponseIndex ([Query Index])
Connection 에 Add 된 Query Index 를 전체 Response Index 로 변환하여 반환한다.
(멀티 Connection 일 때 사용한다.)
[Response Index] – 전체 Connnection Control 에 Add 된 Response Index 를 리턴
[Query Index] – Connection 에 Add 된 Query Index

GetResponseType

GetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 을 반환한다. (다중 IF 연결시 사용)

▶ 함수 정의 및 인자

[Response Type] = GetResponseType ()
Connection 에서 리턴 되는 DataSet 의 DataType 을 반환한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

SetResponseType

SetResponseType 함수는 Connection 에서 리턴되는 DataSet 의 Data Type 의 종류를 설정한다.(다중 IF 연결시 사용)

□ 함수 정의 및 인자



Connection 에서 리턴 되는 DataSet 의 DataType 을 설정한다. (멀티 Connection 일 때 사용한다.)
[Response Type] – DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

AddResponseString

AddResponseString 함수는 TXT DataSet 의 스트링을 추가하는 함수이다.

▶ 함수 정의 및 인자

AddResponseString ([TXT DataSet 스트링])
TXT DataSet 스트링을 추가하는 함수이다
[TXT DataSet 스트링] - 추가하는 TXT DataSet 스트링

SetSeperator

SetSeperator 함수는 Connection 의 Row Seperator 및 Col Seperator 를 추가한다. (웹서버에서 오는 TXT 데이터가 기본 구분자 ('\n', '\t') 를 사용하지 않을 경우 사용한다. Connection 당 한가지만 설정할 수 있다.)

▶ 함수 정의 및 인자

SetSeperator ([Row Seperator],[Col Seperator])
Connection 의 Row, Col Seperator 정보를 추가한다
[Row Seperator] = Connection 의 Row Seperator
[Col Seperator] = Connection 의 Col Seperator

GetRowSeperator

GetRowSeperator 함수는 Connection 의 Row Seperator 를 리턴한다. (DataSet 의 Seperator 설정시 사용)

▶ 함수 정의 및 인자

[Row Seperator] = GetRowSeperator ()
Connection 에 설정된 Row Seperator 를 리턴한다. 특별히 설정하지 않으면 기본값 ('\n')을 리턴한다.



[Row Seperator] – Connection 에 설정된 Row Seperator

${\bf Get Col Seperator}$

GetColSeperator 함수는 Connection 의 Col Seperator 를 리턴한다. (DataSet 의 Seperator 설정시 사용)

□ 함수 정의 및 인자

[Col Seperator] = GetColSeperator ()
Connection 에 설정된 Col Seperator 를 리턴한다. 특별히 설정하지 않으면 기본값 ('\t')을 리턴한다.
[Col Seperator] – Connection 에 설정된 Col Seperator

3.2.9 RQDataSetCtrl

AddDataSet

AddDataSet 함수는 DataSet 을 순서대로 종류에 따라 생성하고, 추가하는 함수이다.

▶ 함수 정의 및 인자

[DataSet 객체] = AddDataSet([DataSet Type])
DataSet 을 Type 에 따라 생성하고 추가하는 함수이다
[DataSet 객체] = 생성된 DataSet
[DataSet Type] - txt : 텍스트 형태(행/열 구분자 포함)
- xml : XML 구조

GetDataSet

GetDataSet 함수는 DataSet 을 추가된 순서의 인덱스에 따라 가져오는 함수이다.



□ 함수 정의 및 인자

[DataSet 객체] = GetDataSet([DataSet Index])
DataSet 객체를 추가된 순서의 인덱스에 따라 반환하는 함수이다.
[DataSet 객체] = DataSet 객체
[DataSet Index] - 가져올 DataSet 의 Index

GetCount

GetCount 함수는 추가된 결과 DataSet 의 갯수를 반환한다.

▶ 함수 정의 및 인자

[DataSet 개수] = GetCount()
추가된 결과 DataSet 의 갯수를 반환한다
[DataSet 개수] = 추가된 DataSet 개수

SetDataSet

SetDataSet 함수는 DataSet 을 입력받은 index 위치에 종류에 따라 생성하고, 추가하는 함수이다.

▶ 함수 정의 및 인자

[DataSet 객체] = SetDataSet([Index], [DataSet Type])
DataSet 을 Type 에 따라 생성하고 해당 Index 에 추가하는 함수이다
[DataSet 객체] = 생성된 DataSet
[Index] - 추가될 위치 인덱스
[DataSet Type] - txt : 텍스트 형태(행/열 구분자 포함)
- xml : XML 구조

3.2.10 RQDataSetTXT

SetDataSetFile

SetDataSetFile 함수는 DataSetFile 의 위치를 설정하는 함수이다.

▶ 함수 정의 및 인자

SetDataSetFile([DataSet File])
DataSet 객체에 DataSetFile 의 위치를 설정하는 함수이다.
[DataSet File] - DataSet 파일 위치

SetEncoding

SetEncoding 함수는 결과 DataSet 의 Encoding 을 설정한다. 하나의 DataSet 에는 Encoding 을 하나만 설정할 수 있다. 따라서 여러 Encoding 을 포함하기 위해서는 여러 개의 DataSet 을 만들어야 한다.

□ 함수 정의 및 인자

SetEncoding([Encoding Type])
결과 DataSet 의 Encoidng 을 설정한다.
[Encoding Type] - Encoding 값
(utf-8, utf-16, ansi)

SetSeperator

SetSeperator 함수는 결과 DataSet 의 Row/Col Seperator 를 설정한다. 하나의 DataSet 에는 Row/Col Seperator 를 하나만 설정할 수 있다.

□ 함수 정의 및 인자

SetSeperator([Row Sep],[Col Sep])
결과 DataSet 의 Row/Col Seperator 를 설정한다.



[Row Sep] - DataSet 의 Row Seperator
[Col Sep] - DataSet 의 Col Seperator

GetRowCount/GetColCount

GetRowCount/GetColCount 함수는 생성된 결과 DataSet Info 의 Row/Col 의 갯수를 반환한다. DataSet Info 는 Viewer Ctrl 의 Run 명령 이후에 DataSet Info 가 생성된다.

□ 함수 정의 및 인자

[DataSet Info Row 개수] = GetRowCount()
[DataSet Info Col 개수] = GetColCount()
결과 DataSet Info 의 Row/Column 갯수를 반환한다.
[DataSet Info Row 개수] = DataSet Row 개수
[DataSet Info Col 개수] = DataSet Col 개수

GetValue

GetValue 함수는 생성된 결과 DataSet Info 의 Row/Col 인덱스에 따른 결과 값을 반환한다. DataSet Info 는 Viewer Ctrl 의 Run 명령 이후에 DataSet Info 가 생성된다.

▶ 함수 정의 및 인자

[DataSet 값] = GetValue([Row 인덱스],[Col 인덱스])
결과 DataSet Info 의 해당 Row/Col 위치의 값을 반환한다.
[DataSet 값] = 해당 DataSet 값
[Row 인덱스] - Row 인덱스 위치
[Col 인덱스] - Col 인덱스 위치

SetDataSetType

SetDataSetType 함수는 결과 DataSet 의 Type 을 설정한다. REQUBE 서버에서 리턴하는 Header 가 있는 TXT DataSet(1: RQCSV)과 Header 가 없는 DataSet(TXT: 3)을 구분하여 설정할 때 사용한다.



□ 함수 정의 및 인자

SetDataSetType([DataSet Type])
결과 DataSet 의 DataSet Type 을 설정한다.
[DataSet Type] - DataSet 의 DataType (1 : RQCSV, 2 : XML, 3: TXT)

3.2.11 RQDataSetXML

SetDataSetFile

SetDataSetFile 함수는 DataSetFile 의 위치와 XML 파일의 반복 Root 를 설정하는 함수이다.

□ 함수 정의 및 인자

SetDataSetFile([DataSet File],[XML Root])
DataSet 객체에 DataSetFile 의 위치와 XML Root 를 설정하는 함수이다.
[DataSet File] - DataSet 파일 위치
[XML Root] - XML Root 위치

SetBindingOption

SetBindingOption 함수는 XML Element 를 추출할 때 Element 이름을 기준으로 추출할지, 순서를 기준으로 추출할지를 설정한다.

▶ 함수 정의 및 인자

SetBindingOption([Binding Option])
XML Element .Binding 을 이름기준으로 할지, Element 순서대로 할지 설정하는 함수
[Binding Option] - Binding Optioin (이름기준 : 1, 순서기준 : 2)

3.3 API

다음은 사용자의 편의를 위해 API 함수를 객체별로 분류하여 오름차순 인덱스화하여 수록하였다.

뷰어(RQViewerCtrl)

BackColor	. 5 4
ButtonMoveFirstPageClickAfter 이벤트	. 7 1
ButtonMoveFirstPageClickBefore 이벤트	. 7 0
ButtonMoveLastPageClickAfter 이벤트	. 7 2
ButtonMoveLastPageClickBefore 이벤트	. 7 1
ButtonMoveNextPageClickAfter 이벤트	. 6 9
ButtonMoveNextPageClickBefore 이벤트	. 6 8
ButtonMovePrevPageClickAfter 이벤트	. 6 8
ButtonMovePrevPageClickBefore 이벤트	. 6 7
ButtonPrintClickAfter 이벤트	. 6 1
ButtonPrintClickBefore 이벤트	. 6 0
ButtonRefreshClickAfter 이벤트	. 6 2
ButtonRefreshClickBefore 이벤트	. 6 1
ButtonSaveAsClickAfter 이벤트	. 6 0
ButtonSaveAsClickBefore 이벤트	. 5 9
ButtonStopClickAfter 이벤트	. 6 4
ButtonStopClickBefore 이벤트	. 6 3
ButtonZoomInClickAfter 이벤트	. 6 6
ButtonZoomInClickBefore 이벤트	. 6 6
ButtonZoomOutClickAfter 이벤트	. 6 5
ButtonZoomOutClickBefore 이벤트	. 6 4
CloseReport	. 3 9
CreateConnectionControl	. 3 7
CreateDataSetControl	. 3 8
CrossTabPageBreak	. 5 2
EndRunReport 이벤트	. 5 7
EmptyDataCheckOption	. 5 6
EmptyResultData 이벤트	. 5 7
GetConnectionControl 함수	. 3 8

GetDataSetControl 함	<u> </u>	9
GetReport		В
GetDocType	5 3	3
GetTotalPageNo	5 3	3
GetCurrentPageNo	5 4	4
MoveFirtstPage	4 C	0
MoveLastPage	4 1	1
MoveNextPage	4 1	1
MovePage	4 2	2
MovePrevPage	4 2	2
OpenReport	3 6	6
Print	4 7	7
PrintDialog	4 8	8
PrintPreview	4 9	9
Refresh 이벤트	5 ε	В
Refresh	4 C	O
Run		7
ShowProgressDialog.	5 €	6
ShowToolBarButton	4 5	5
StatusBarVisible	5 5	5
Stop	3 9	9
SaveAsDialog	4 S	9
SaveAs	5 C	O
SendToExcel	5 C	0
SendToPDF	5 1	1
SetBaseURL	5 2	2
ToolBarVisible	5 5	5
Zoom	4 4	4
ZoomIn	4 4	4
ZoomOut	4 3	3
리포트 객체(RQReport	t)	
GetObject	7 3	3
GetRuntimeVariable	7 3	3
GetSQLControl	7 4	4
SetRuntimeVariable	7 3	3



리포트 컴포넌트 객체(RQObject)	
GetProperty	
SetProperty	7 4
SQL 컨트롤 객체(RQSQLControl)	
GetConnectionString	
GetCount	
GetDBID	
GetDBString	
GetDefaultRowSeperator / GetDefaultColSeperator	
GetQueryCount	
GetSQLString	
GetXMLRoot	7 7
GetDBIFType	7 7
GetDBInfo	
GetSQLStmt	
GetSQLStmtCount	
GetRowSeperator	7 9
GetColSeperator	7 9
GetEncoding	8 0
RQConnectionCtrl 객체	
AddConnection	86
GetConnection	87
GetCount	89
GetConnectionByName	
GetConnectionByID	
GetConnectionType	
Send	
RQConnectionHTTP 객체	
AddParameter	
GetCount	1 0 2
GetEncoding	1 0 1
GetParameter	
GetPath	99

	RemoveParameter	1 (0 .	1
	ReplaceParameter	1 (0 :	2
	SetEncoding	1 (0 :	3
	SetPath	!	9 !	9
	SetQueryCount	1 (0 :	2
	GetURL	1	0 ;	3
	AddResponseCount	1	0 ;	3
	GetResponseIndex	1 (0 4	4
	GetResponseType	1 (0 4	4
	SetResponseType	1 (0 4	4
	SetXMLRoot	1 (0 :	5
	GetXMLRoot	1 (0 :	5
	SetSeperator	1 (0 :	5
	GetRowSeperator	1 (0 (6
	GetColSeperator	1 /	0 (6
R	QConnectionJDBC 객체			
	AddDBInfo	1	1 (0
	AddSQLStmt	1	1	1
	AddResponseCount	1	1	1
	GetCount	1 /	0	7
	GetEncoding	1 /	0	7
	GetResponseFile	1 /	0	7
	GetResponseIndex	1 /	0 !	9
	GetResponseType	1	1 (0
	Response	1	0 (6
	SetDBString	1 /	0 8	8
	SetQueryCount	1	0 8	8
	SetSQLStmt	1	0 !	9
	SetQueryString	1	0 8	8
	SetDBInfo	1	0 !	9
	SetResponseType	1	1 (0
R	QConnectionLOCAL 객체			
	AddResponseFile	1	1 8	8
	AddXMLResponseFile	1	1 !	9
	AddXMLResponseString	1	1 9	9

AddResponseString
GetCount
GetEncoding
GetResponseFile
GetResponseIndex
GetResponseType
GetXMLRoot
GetRowSeperator
GetColSeperator
SetEncoding
SetResponseType
SetSeperator
RQDataSetCtrl 객체
AddDataSet
GetCount
GetDataSet
SetDataSet
RQDataSetTXT 객체
GetRowCount/GetColCount
GetValue
SetDataSetFile
SetEncoding
SetSeperator
SetDataSetType
RQDataSetXML 객체
SetDataSetFile
CatDindingOntion 1.0.6