AppBuilder
By Magic Software Enterprises

Magic Software AppBuilder

Version 3.2

IVP User Guide

Corporate Headquarters:

Magic Software Enterprises
5 Haplada Street,
Or Yehuda 60218, Israel
Tel +972 3 5389213
Fax +972 3 5389333

IVP User Guide

Introduction to IVP

The Installation Verification Program (IVP) is an application designed to help you verify your AppBuilder installation and its configuration for use with third-party programs.
IVP comes with AppBuilder 2.0.3.5 Host and AppBuilder 3.1n. It is only supported at these levels and the third-party matrix associated with these levels.
This document provides instructions for you to import IVP, create the necessary database tables, and prepare the application. This document also provides instructions for you to use to start the IVP application and perform the various tests that IVP can perform.

Testing Scenarios

IVP is designed to conduct tests on client, server, and mainframe environments. Tests for all these environments can be run from the IVP Command window.
IVP in its entirety tests more than 80 operations in AppBuilder. You probably need only a subset of the number of tests that IVP can perform. The tests that you perform using IVP depend on the environment you are testing and using at your site.

> **This manual presumes some familiarity with AppBuilder and its documentation package (including the Glossary of terms in _Getting Started Guide**

Client Tests

If you have configured AppBuilder to prepare local standalone applications on a client machine, you can run the client verifications. For instructions on performing client tests, see Client Verifications. In client environments, IVP conducts the following tests:

- C Client Calling C User Components
- C Client Performing Database Calls
- Java Client Calling C User Component
- Java Client Calling Java User Components
- Java Client Performing Database Calls
- Java Reports with Java Runtime

Client/Server Tests

If you have configured AppBuilder to prepare applications on an EJB or Webservices server, you will run server verifications. For instructions to perform server tests, see Server Verifications.
In EJB and Web services server environments, IVP conducts the following tests:

- EJB and Webservices Rule Test
- EJB and Webservices Database Rule Test
- Rule Test by way of a Gateway
- Database Rule by way of a Gateway
- EJB and Webservices C Component Test
- C Component Test by way of a Gateway
- EJB and Webservices C Database Component Test
- C Database Component Test by way of a Gateway
- Java Component Test
- Java Component Test by way of a Gateway
- Java Database Component Test
- Java Database Component Test by way of a Gateway

If you have configured AppBuilder to prepare applications on a C server, you will run server verifications. For instructions to perform server tests, see C Server Tests.
In C server environments, IVP conducts the following tests:

- C Server Rule Test
- C Server Database Rule Test
- C Server C Component Test
- C Server C Database Component Test

In addition, IVP includes Servlet Tests.

Mainframe

In mainframe environments, the Installation Verification Program can be used to test both the AB3.1 and the AB2035 installed components. For instructions on performing tests on the mainframe, see Mainframe Verifications. IVP conducts the following tests in the following environments:

CICS

If you have configured AppBuilder to run under CICS, you can run verifications on the mainframe. For instructions on performing CICS environment tests, see Verifying the CICS Environment.

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- C Component Test
- C Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test

CICS/BATCH

If you have configured AppBuilder to run under CICS/BATCH, you can run verifications on the mainframe. For instructions on performing CICS/BATCH environment tests, see Verifying the CICS/BATCH Environment.

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test

PCCICS

If you have configured AppBuilder to run under PCCICS, you can run verifications on the workstation. For instructions on performing PCCICS environment tests, see Verifying the PCCICS Environment.

- Mainframe Rule Test

BATCH

If you have configured AppBuilder to run under BATCH, you can run verifications on the mainframe. For instructions on performing BATCH environment tests, see Verifying the Batch Environment.

- ClassicCOBOL Tests
- ClassicCOBOL DB2 Tests
- OpenCOBOL Tests
- OpenCOBOL DB2 Tests
- Batch Report Test

IMS

If you have configured AppBuilder to run under IMS, you can run verifications for that mainframe environment. The application must be prepared on the mainframe but can be run from a workstation. For instructions on performing IMS environment tests, see Verifying the IMS Environment.

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test
- C Component Test
- C Database Component Test

PCIMS

If you have configured AppBuilder to run under PCIMS, you can run verifications on the workstation. For instructions on performing PCIMS

environment tests, see Verifying the PCIMS Environment.

- Mainframe Rule Test

Requirements

To run the Installation Verification Program, you must have installed and configured AppBuilder 3.1 for the PC or AppBuilder 2.0.3.5 for the mainframe or both. Those versions of AppBuilder have other requirements listed in their respective installation manuals and the Third-Party Support Matrix. In general, however, if you are going to build and deploy applications built in C for Windows, you must have a supported C compiler. Likewise, if you are building applications for Java deployment, you must have a supported Java compiler. For mainframe testing, you must have the appropriate mainframe compilers to run programs on the mainframe. For details about supported compiler versions, see the Third-Party Support Matrix. All versions of AppBuilder require some form of database for its repository, and the supported kinds and versions are also listed in the Third-Party Support Matrix.

IVP Migration and Setup

To use the Installation Verification Program (IVP), you must migrate it to your repository, set up database tables, and prepare the components that you are configuring. Database tables do not need to be set up on the mainframe since the application uses existing IBM tables.
This section discusses the steps necessary to run IVP and focuses on the following steps:

- Importing IVP
- Setting Up the Database Tables

Importing IVP

IVP requires default objects to be in the personal or workgroup repository on a personal computer or workstation. To do this, you should migrate the Installation Verification Program into a clean repository populated with the default repository.
The steps necessary to import IVP to a clean Personal Repository differ from the steps necessary to import IVP to a clean Workgroup Repository.

Migrating IVP to a Personal Repository

To migrate the repository with IVP to a clean Personal Repository, complete the following steps:

1. Create a Personal Repository for use with IVP.
    - From the Start menu, select AppBuilder > Repository > Repository Administration. The Repository Administration tool appears.
    - In the Repository Administration dialog select Repository > New > Repository. The Repository Setup Wizard displays.
    - Select the DBMS that you are using with AppBuilder and click Next.
    - Type the name of the new repository.
    - Enter your user ID and password.
    - (Optional) Specify the working directory for the repository.
    - Click Next.
    - Specify the location for the new repository and click Next.
    - Enter the host connection parameters and click *Next* .
    - The Confirm Installation window displays.
    - Click Next or Create.
      For more details about the Repository Administration tool, see the *Repository Administration Guide for Workgroup and Personal Repositories* .
2. Use the Repository Administration tool to connect to the Repository you have created.
    - Select Repository > Connect.*
    - Choose the Personal Repository alias.
    - Type your user ID and password and click Next.
3. From *Tools* on the Repository Administration menu bar, select *Migration Import* > *Open* . The Migration Import dialog appears.
4. Select *File* > *Choose Directory* .
5. Choose the IVP directory in the AppBuilder installation folder and click OK.
6. Select *Action* > *Analyze* .
   An analysis is performed on the migration files in the selected directory.
   Click the Summary tab and make sure that the analyze was successful.
7. Select *Action* > *Import* .
   An Active Unit of Work dialog saying that the repository is using a Unit of Work appears.
   A Unit of Work (UOW) is a facility that tracks your changes for ease in upload and download and migration export.
8. If you want to add the Installation Verification Program objects to the Unit of Work, click *Yes* . Otherwise, click *No* .
   The results of the migration show in the Import tab. Check the Errors column on the Import tab to make sure that the import is successful.
   The import is successful if the log file displays a statement of successful with no unsuccessful messages.

Migrating IVP to a Workgroup Repository

To migrate the repository with IVP to a clean workgroup repository, complete the following steps:

1. Use the Repository Administration tool to connect to the Workgroup Repository.
    - Select Repository > Connect.
    - Choose the Workgroup Repository alias.
    - Type your user ID and password and click Next.
      For more details about the Repository Administration tool, see *Repository Administration Guide for Workgroup and Personal Repositories* .
2. From *Tools* on the Construction Workbench menu bar, select *Migration Import* .
3. Select *File* > *Choose Directory* .
4. Choose the IVP directory in the AppBuilder installation folder and click OK.
5. Select *Action* > *Analyze* .
   An analysis is performed on the files in the selected migration object in the directory.
   Click the Summary tab and make sure that the analyze was successful.
6. Select *Action* > *Import* .
   An Active Unit of Work dialog saying that the repository is using a Unit of Work appears.
   A Unit of Work (UOW) is a facility that tracks your changes for ease in upload and download and migration export.
7. To add IVP objects to the Unit of Work, click *Yes* . Otherwise, click *No* .
   The results of this action show in the Import tab. Check the Errors column to make sure that the action is successful. The action is successful if the log file displays a statement of successful with no unsuccessful messages.

Setting Up the Database Tables

Before you Prepare the Installation Verification Program, you must set up database tables so that client or server database tables can be tested at runtime.
This section discusses the following topics:

- Setting Up the Client Database Tables
- Setting Up the Server Database Tables

> ⚠ **Mainframe environments have their own database tables supplied by default through IBM (such as SYSIBM.SYSTABLES). You do not have to set up database tables for mainframe environment runtime.**

Setting Up the Client Database Tables

To set up the client database tables for IVP database tests, complete the following steps:

1. In Construction Workbench, select File > Open Project
2. Click Query.
3. Insert IVP_FUNCTION in the Query Value field and click *Open* .
4. Select *File* > *Project Options* .
5. Fill in *Application options* for the database to be used with IVP.
6. When all database setting changes have been made, click *OK* .
7. Click the Configuration tab.
8. Expand the client partition to which you want to prepare IVP.
9. Edit the Database object properties for IVP_LOCAL_DB, including the database implementation name, to match those set up in the project.
10. Expand the Database object.
11. Prepare all files under the Database Object.
    - For C, select the File Object IVP_DB_FLE and then select Build > Prepare.
    - For Java, prepare the File Objects IVP_DB_FLE and IVP_JAVA_RPT_EMP_FLE.
12. Select Build > Create Table for the file or files that you prepared.

Setting Up the Server Database Tables

To set up the server database tables for Installation Verification Program database tests, complete the following steps:

1. Expand the server partition to which you want to prepare the Installation Verification Program.*
2. Edit the Database object properties for IVP_DB, including the database implementation name.
3. Expand the Database object.
4. Prepare all files under the Database Object.
    - For C, select the File Object IVP_DB_FLE and then select Build > Prepare.
    - For Java, prepare the File Objects IVP_DB_FLE and IVP_JAVA_RPT_EMP_FLE.
5. Select Build > Create Table for the file or files that you prepared.

Client Verifications

Client tests verify configurations for local standalone applications. You can run a client runtime test using AppBuilder Rule source generated as C

(the C client partition) or AppBuilder Rule source generated as Java (the Java client partition).
A *partition* is an object in AppBuilder that defines that a segment of an application is to be prepared, or built on, a specific machine or platform type. For more details about configurations of distributed applications, see the *Deploying Applications Guide* . In the case of the IVP application, it has already been built with multiple partitions for appropriate environments. You do not have to modify properties except as specified below.
Using AppBuilder Rule source generated as C, you can verify a configuration to call a native C user component and you can run SQL against a local database configuration. See:

- Starting IVP for C
- Verifying C Runtime

Using AppBuilder Rule source generated as Java, you can verify configurations to call a native C user component and a Java user component. You can also run SQL against a local database configuration and generate a Java Report to be sent to a printer. See:

- Starting IVP for Java
- Verifying Java Runtime

Configuring the C Client Partition

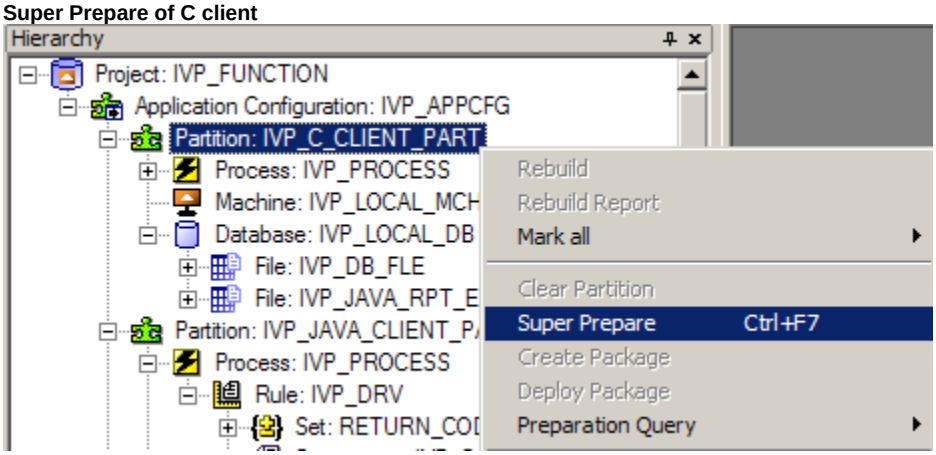
You must configure your machine so that the local account must have database access. Specifically, you must have the right to insert, select, and delete objects in the database for the client database test to operate as designed. The details on this database configuration depend, to some extent, on the operating system and database you use. If you have questions, see your database administrator.

Super Preparing the C Client Partition

This section explains how to super prepare the C Client partition.

> ⚠ **To prepare the C client partition, you must have a supported C compiler installed and configured.**

Complete the following steps:

1. Select the C client partition.
   Super Prepare of C client
   

2. Use the right-click menu and select *Super Prepare* or select *Build* > *Super Prepare* .
   All object prepares should be successful.
   Output from Super Prepare

Name	Type	Submission Status	Submission Time	Last Modified
IVP_JAVA_RPT_EMP_FLE	FILE	Successful	11/16/2006 - 11:40...	11/4/2006 - 9:51:47 AM
IVP_DB_FLE	FILE	Successful	11/16/2006 - 11:40...	11/10/2006 - 10:47:11 PM
IVP_JAVA_RPT_DB	RULE	Successful	11/16/2006 - 11:40...	11/4/2006 - 9:51:47 AM
IVP_BMP	BITMAP	Successful	11/16/2006 - 11:40...	11/4/2006 - 9:51:30 AM
RETURN_CODES	SET	Successful	11/16/2006 - 11:40...	11/18/2005 - 3:36:42 PM
IVP_COMMAND_WND	WINDOW	Successful	11/16/2006 - 11:40...	11/4/2006 - 9:51:29 AM
IVP_C_CLT	COMPONENT	Successful	11/16/2006 - 11:40...	11/4/2006 - 9:51:30 AM
IVP_CLT_DB	RULE	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:31 AM
IVP_SET_AUTHENTICATION	COMPONENT	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:31 AM
IVP_JAVA_RPT_RLE	RULE	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:44 AM
IVP_MVS_CLT_DRV	RULE	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:36 AM
IVP_DRV	RULE	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:29 AM
IVP_FUNCTION	FUNCTION	Successful	11/16/2006 - 11:41...	11/4/2006 - 9:51:26 AM

Configuring the Java Client Partition

Configuring the Java client partition is similar to Configuring the C Client Partition. The database object needs to have the database name, and the Project Options need to contain the database, user ID, password, and database type.

Super Preparing the Java Client Partition

This section explains how to super prepare the Java Client partition.

> **To use AppBuilder Rules to verify Java client configurations, you must have a supported Java compiler installed on the client machine.**

Use the procedure specified for Super Preparing the C Client Partition and select the IVP_JAVA_CLIENT_PART.

Output from Java client Super Prepare

Name	Type	Submission Status	Submission Time	Last Modified
IVP_JAVA_CLIENT_PART	PARTITION	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:52 AM
IVP_JAVA_RPT_EMP_FLE	FILE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:47 AM
IVP_DB_FLE	FILE	Successful	11/16/2006 - 11:48...	11/10/2006 - 10:47:11 PM
IVP_JAVA_RPT_DB	RULE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:47 AM
IVP_BMP	BITMAP	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:30 AM
IVP_COMMAND_WND	WINDOW	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:29 AM
IVP_C_CLT	COMPONENT	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:30 AM
IVP_JAVA_CLT	COMPONENT	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:30 AM
IVP_CLT_DB	RULE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:31 AM
IVP_SET_AUTHENTICATION	COMPONENT	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:31 AM
IVP_JAVA_RPT_RLE	RULE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:44 AM
IVP_MVS_CLT_DRV	RULE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:36 AM
IVP_DRV	RULE	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:29 AM
IVP_JAVA_CLIENT_PART	PARTITION	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:52 AM
IVP_FUNCTION	FUNCTION	Successful	11/16/2006 - 11:48...	11/4/2006 - 9:51:26 AM

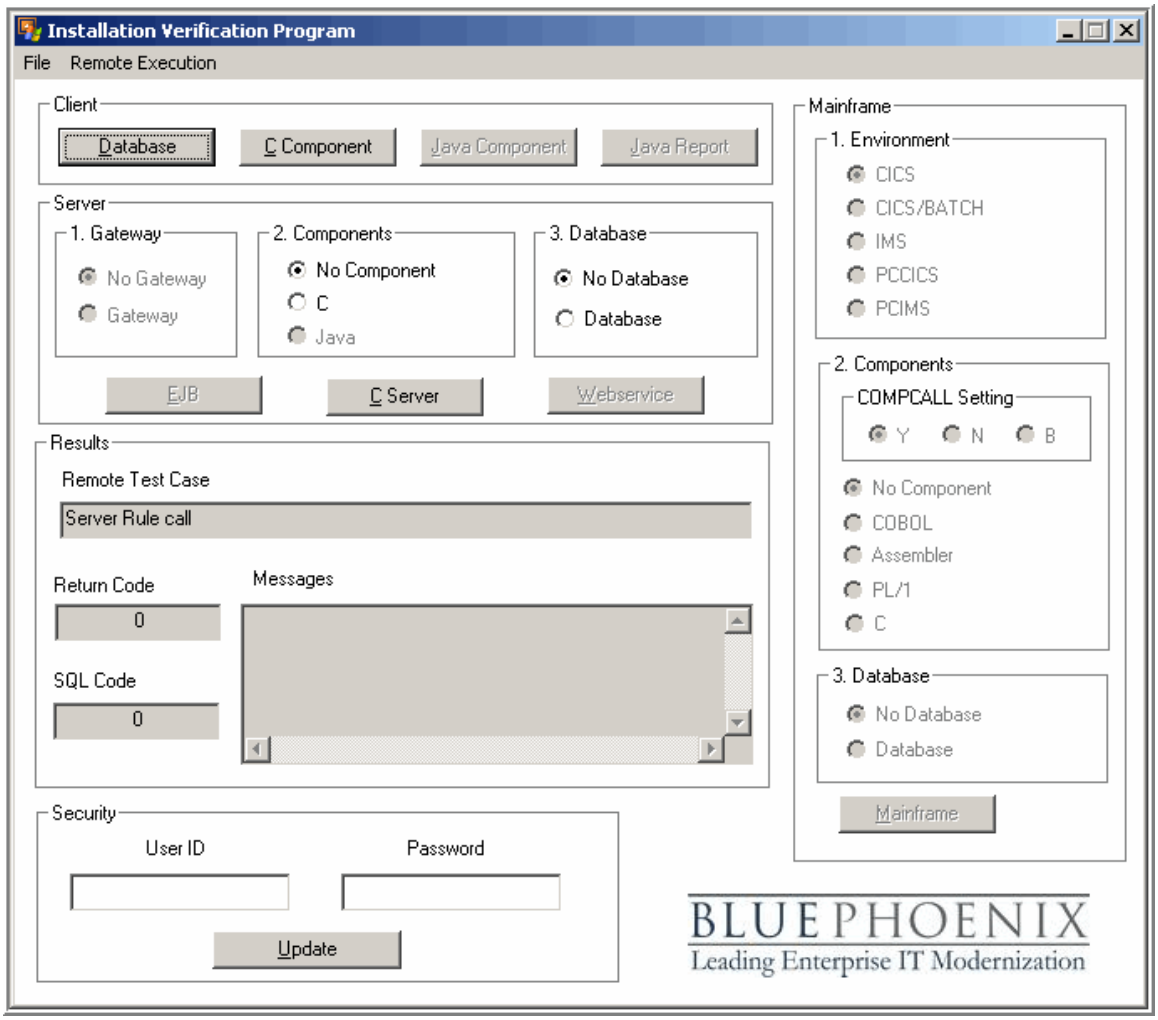
Starting IVP for C

The Installation Verification Program (IVP) works from a single window, called the IVP Command Window. There are two graphical interfaces: the Windows interface and the Java interface. To start IVP for C, complete the following steps:

1. Select Run > Windows.
   A menu with IVP_FUNCTION displays.
2. Select *IVP_FUNCTION* > *IVP_PROCESS* .
   A dialog asking if you want to run RuleView displays.
3. If you want to run RuleView, click *Yes* . Otherwise, click *No* .

The IVP C Client Command Window displays.

IVP Command Window for C



Verifying C Runtime

You can use AppBuilder Rules to verify C user components and other generated C applications that access databases using SQL. You can test the following configurations with the C client in IVP:

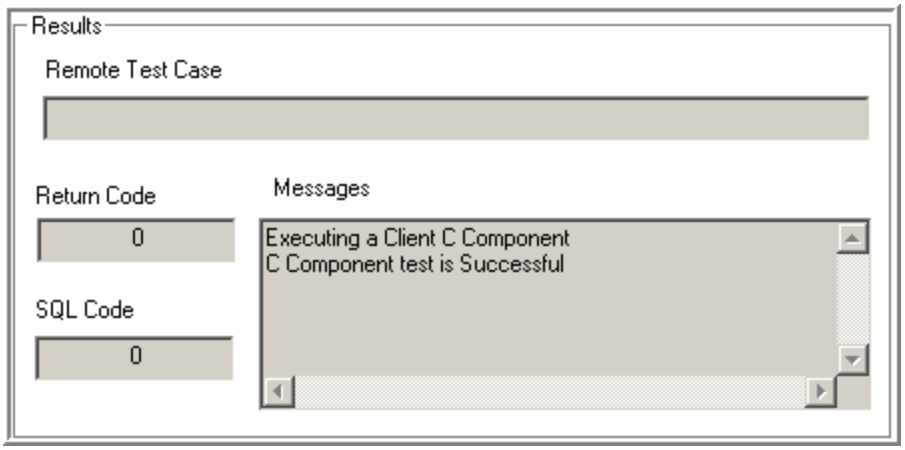
- C Client Calling C User Components
- C Client Performing Database Calls

C Client Calling C User Components

This test verifies that native C is configured properly with AppBuilder.
To test C User Components, complete the following steps:

1. On the IVP Command Window, click ***C Component*** **in the Client section** ***(see IVP Command Window for C) .***
2. Check the Results group box for the results of the test. If the test is successful, the following message displays:

Results calling C User component



C Client Performing Database Calls

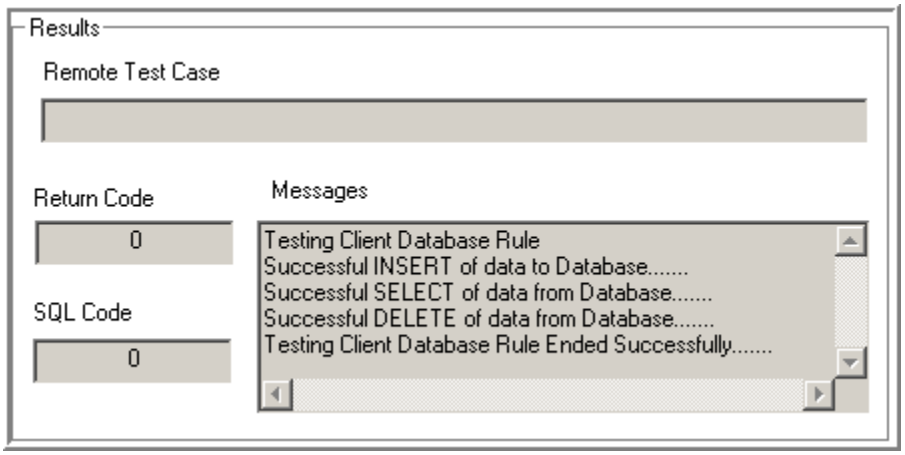
This section explains the database calls performed by the C Client.

> ⚠ **The tester should have *insert* , *select,* and *delete* authority on the database table.**

To test the database configuration, complete the following steps:

1. On the IVP Command Window, click the ***Database*** **button in the Client section (see IVP Command Window for C).**
2. Check the Results group box for the results of the test. If the test is successful, the following message displays:

Results from C performing database calls

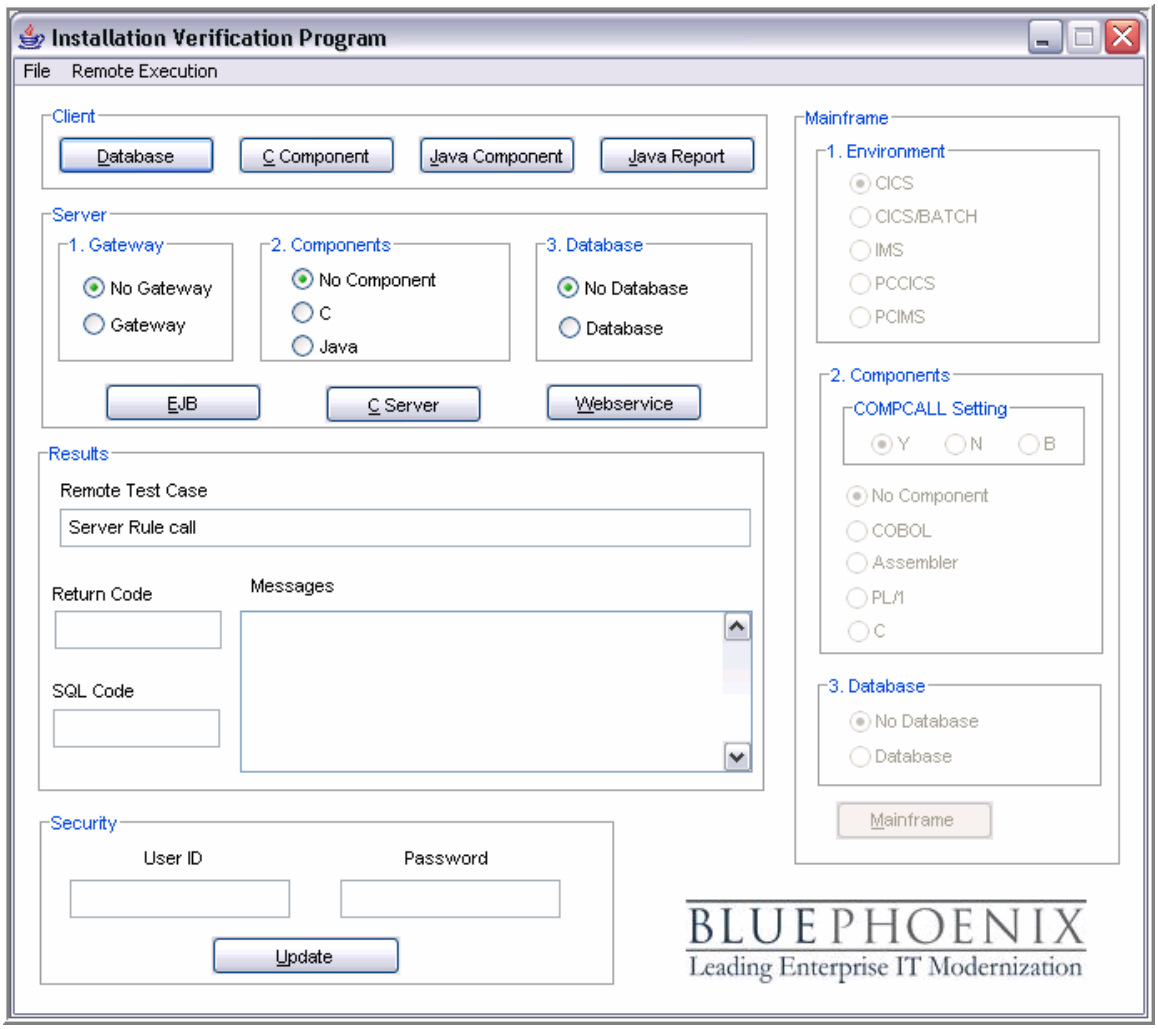


Starting IVP for Java

To start IVP for Java, complete the following steps:

1. Select ***Run > Java*** **.**
   A command window appears. Then a menu with IVP_FUNCTION displays.
2. Select *IVP_FUNCTION > IVP_PROCESS* .
   A dialog asking if you want to run RuleView displays.
3. If you want to run RuleView, click *Yes* . Otherwise, click *No* .
   The IVP Java Command Window displays.

IVP Command Window for Java



Verifying Java Runtime

You can use AppBuilder Rules to verify Java User Components and Rule Source generated for Java that executes SQL commands. You can test the following configurations with Java in IVP:

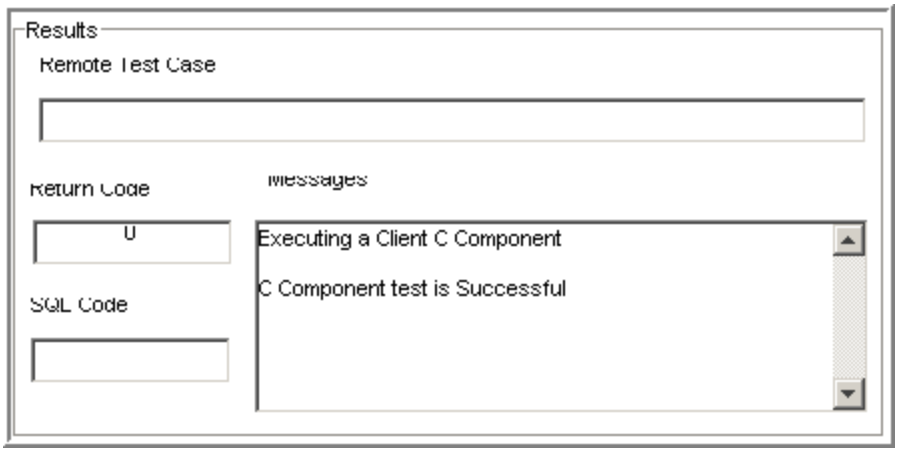
- Java Client Calling C User Component
- Java Client Calling Java User Components
- Java Client Performing Database Calls
- Java Reports with Java Runtime

Java Client Calling C User Component

This test verifies the call of a native C User Component from an AppBuilder generated Java Rule. To get the Java client to find the C User Component, Java must identify the DLL created in the Super Prepare. If AppBuilder is installed in the default C:\Appbuilder directory, look in C:\AppBuilder\JAVA\rt\ivp_appcfg\ivp_java_client_part\BIN for the file IVP_C_CLT.DLL. This file must be available to the Java client. One method of insuring this is to add the path to the machine's PATH environment variable. Another method is to copy the file to a location already defined in the PATH environment variable.
To test the C user component written in Rules Language generated to Java, complete the following steps:

1. In the Java IVP Command Window, click the ***C Component*** **button.**
2. Check the Results group box for the results of the test. The following message displays:

C User Component called from Java rule

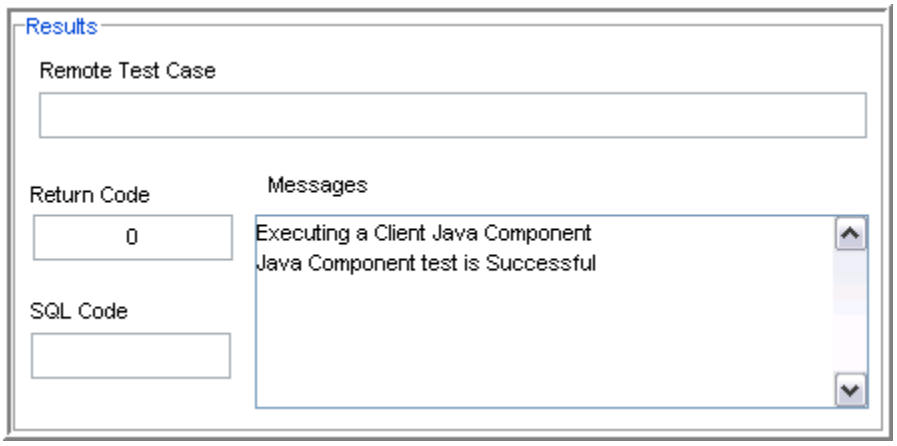


Java Client Calling Java User Components

This test verifies the call of a native Java User Component from an AppBuilder generated Java Rule.
To test Java Components, complete the following steps:

1. Click the ***Java Component*** **button.**
2. Check the Results group box for the results of the test. If the test is successful, the following message displays:

Java User Component



Java Client Performing Database Calls

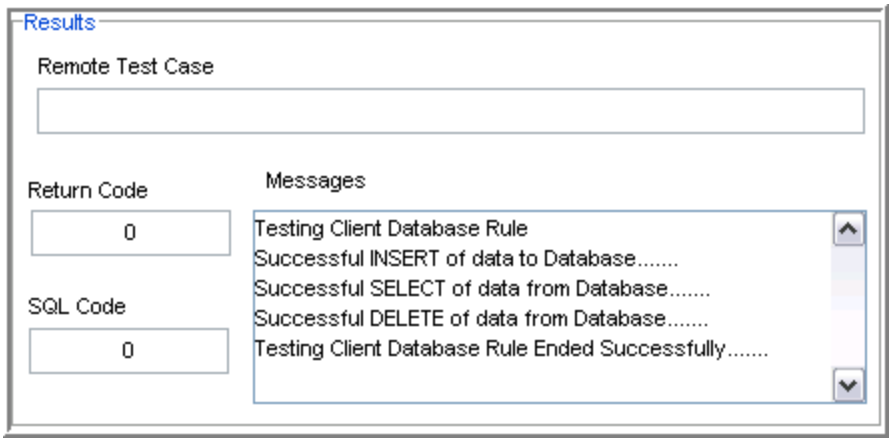
This section explains the database calls performed by the Java Client.

> ⚠ **The tester should have *insert* , *select,* and *delete* authority on the database table.**

To test the database configuration with database SQL, complete the following steps:

1. Click the ***Database*** **button.**
2. Check the Results group box for the results of the test. If the test is successful, the following message displays:

Java client performing database calls



Java Reports with Java Runtime

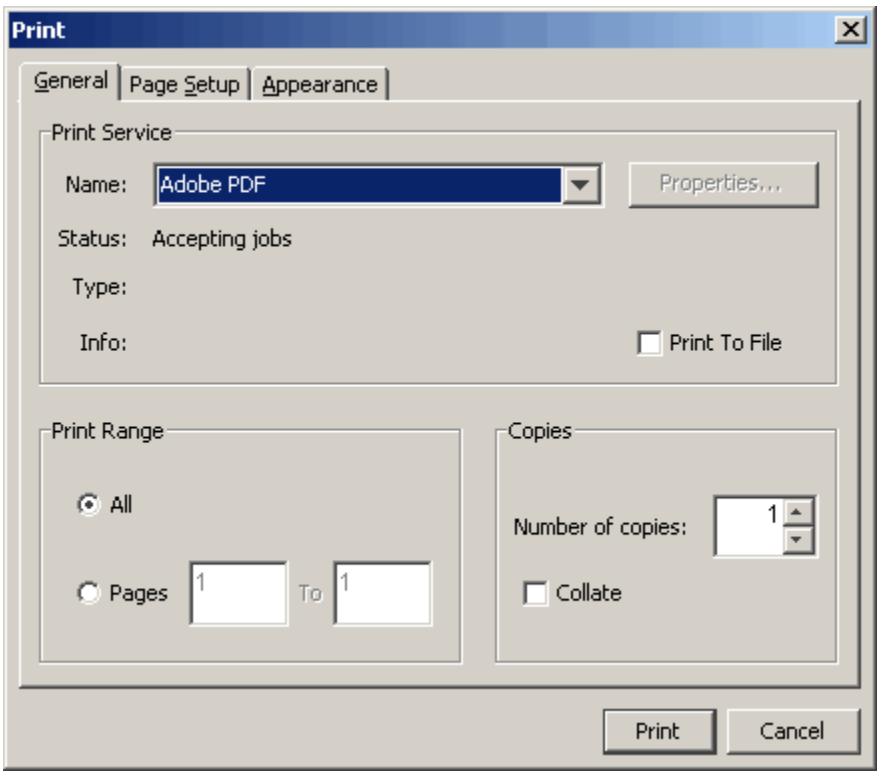
This section explains how to generate Java reports with Java runtime.

> ⚠ **The tester should have *insert* , *select,* and *delete* authority on the database table.**

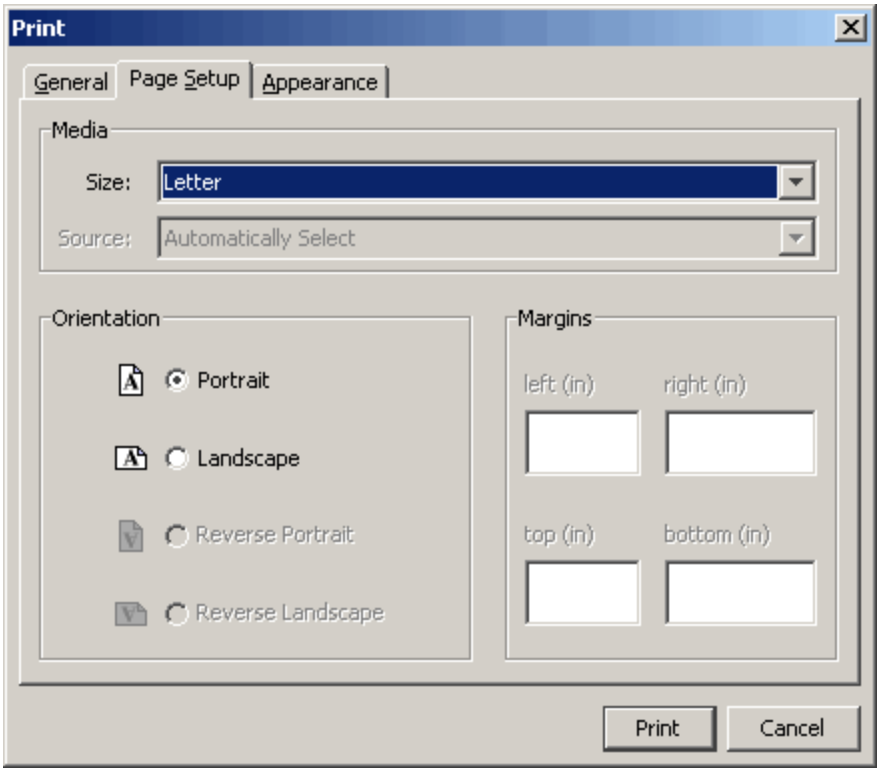
This test verifies the Java report from an AppBuilder generated Java Rule.
To test Java Components written in AppBuilder Rule source generated to Java, complete the following steps:

1. Click the ***Java Report*** **button.** If the test is successful, the Print dialog for your workstation appears:

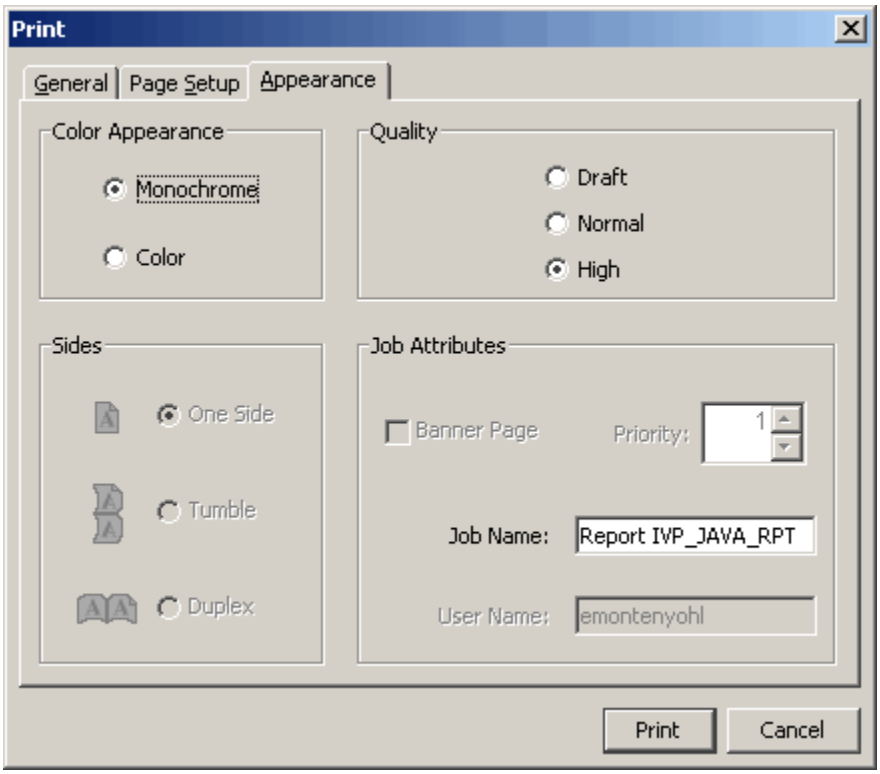
Java print report dialog, General tab



Java print report dialog, Page Setup tab



Java print report dialog, Appearance tab



> ⚠ **The printer list depends upon those installed on the local machine.**

The output of the Java report is 6 pages long and should display like the following:

Java report output, page 1

Week Ending 03/05/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	40
5367421	Debra White	53
7654222	Martin Howard	52
7654321	Sam Miller	40
9987612	Rodney Banks	43

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	40

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	42
1234567	Joe Smith	40
1299872	Matthew Parker	50
3564778	Samantha Ty	47

Total Hours for the Week 490

Week Ending 03/12/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	41
5367421	Debra White	48
7654222	Martin Howard	44
7654321	Sam Miller	41
9987612	Rodney Banks	48

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43

4352616	Sarah Thompson	47

Department **Manager**
Marketing Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	42
1234567	Joe Smith	41
1299872	Matthew Parker	41
3564778	Samantha Ty	45

Total Hours for the Week 481

Week Ending 03/19/05

Department **Manager**
Finance Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	40
5367421	Debra White	43
7654222	Martin Howard	49
7654321	Sam Miller	40
9987612	Rodney Banks	40

Department **Manager**
Human Resources Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	40
4352616	Sarah Thompson	42

Department **Manager**
Marketing Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	41
1234567	Joe Smith	40
1299872	Matthew Parker	40
3564778	Samantha Ty	45

Total Hours for the Week 460

Week Ending 03/26/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	40
5367421	Debra White	46
7654222	Martin Howard	46
7654321	Sam Miller	40
9987612	Rodney Banks	48

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	41

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	42
1234567	Joe Smith	40
1299872	Matthew Parker	40
3564778	Samantha Ty	45

Total Hours for the Week 471

Week Ending 04/02/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	40
5367421	Debra White	44
7654222	Martin Howard	52
7654321	Sam Miller	40
9987612	Rodney Banks	43

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	44
4352616	Sarah Thompson	44

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	41
1234567	Joe Smith	40
1299872	Matthew Parker	40
3564778	Samantha Ty	45

Total Hours for the Week 473

Week Ending 04/09/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	41
5367421	Debra White	42
7654222	Martin Howard	52
7654321	Sam Miller	41
9987612	Rodney Banks	41

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	42

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	42
1234567	Joe Smith	41
1299872	Matthew Parker	49
3564778	Samantha Ty	45

Total Hours for the Week 479

Week Ending 04/16/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	45
5367421	Debra White	45
7654222	Martin Howard	45
7654321	Sam Miller	45
9987612	Rodney Banks	40

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	40

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	41
1234567	Joe Smith	45
1299872	Matthew Parker	41
3564778	Samantha Ty	48

Total Hours for the Week 478

Week Ending 04/23/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	45
5367421	Debra White	75
7654222	Martin Howard	52
7654321	Sam Miller	65
9987612	Rodney Banks	64

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	40

Employee Hours Report 11/16/2006 12:35 PM

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	42
1234567	Joe Smith	65
1299872	Matthew Parker	35
3564778	Samantha Ty	61

Total Hours for the Week 587

Week Ending 04/30/05

Department	Manager
Finance	Martin Howard

Employee Id	Employee Name	Hours
3421265	Amber Williams	45
5367421	Debra White	57
7654222	Martin Howard	52
7654321	Sam Miller	55
9987612	Rodney Banks	45

Department	Manager
Human Resources	Charlene Devoe

Employee Id	Employee Name	Hours
3421234	Charlene Devoe	43
4352616	Sarah Thompson	40

Department	Manager
Marketing	Dan Fields

Employee Id	Employee Name	Hours
1233221	Dan Fields	41
1234567	Joe Smith	55
1299872	Matthew Parker	51
3564778	Samantha Ty	51

Total Hours for the Week 535

Page 6

Server Verifications

The IVProgram can perform a number of verifications of server configurations. There are three main server verifications available:

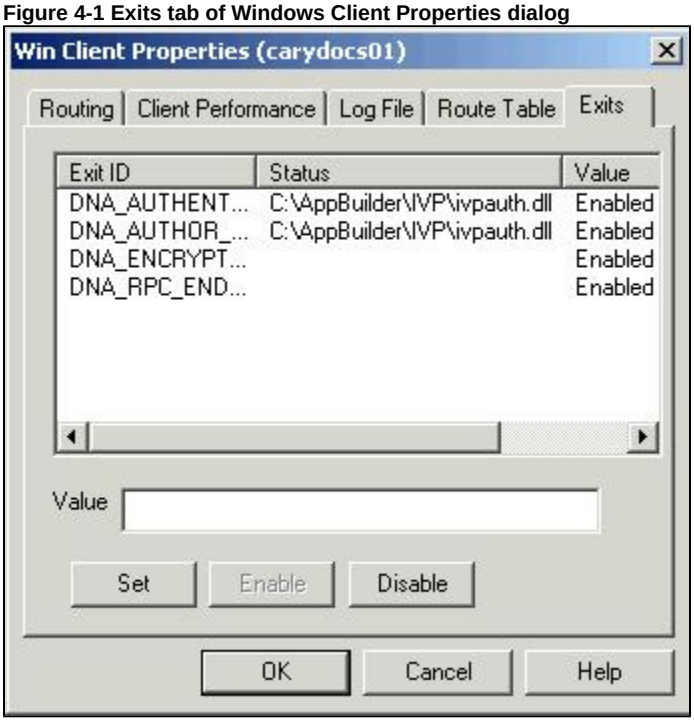
- C Server Tests
- EJB and Webservices Tests
- Servlet Tests

Tip
For more information on configuring, preparing, and deploying applications, see the AppBuilder *Deploying Applications Guide* .

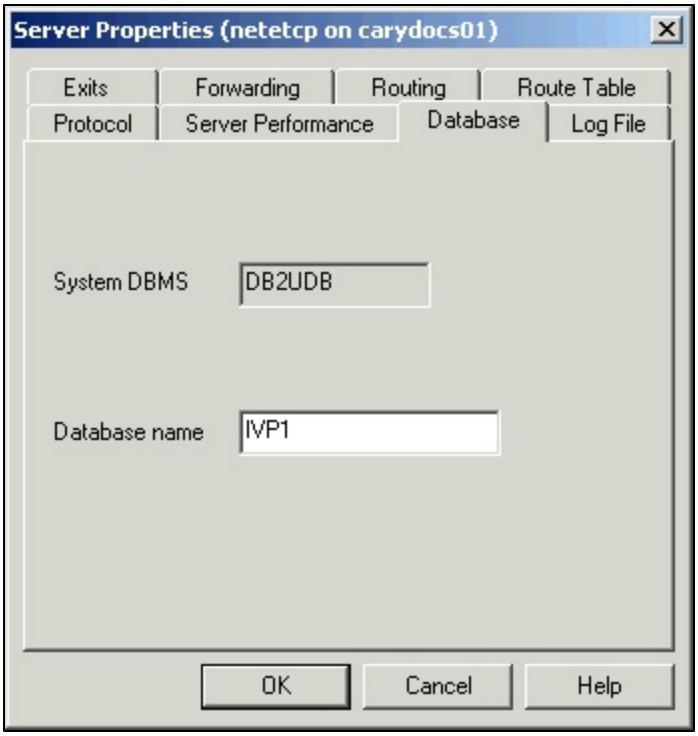
Configuring the Server Partitions

To configure the server partitions, you must specify the authentication exit for the application to use. To do so, complete the following steps:

1. Open the AppBuilder Management Console (Select **Start > Programs > AppBuilder > Configuration > Management Console.**)
2. Right-click the Windows client and select *Properties* . The Server Properties dialog appears.
3. Select the Exits tab. To deploy as a client/server application, the server must have exits set. For IVP, Magic Software provides an exit which enables a user to specify username and password information for a remote machine (server) dynamically. This exit is named *ivpauth.dll* and is located in the IVP directory.
   Figure 4-1 Exits tab of Windows Client Properties dialog



4. Select the DNA_AUTHENT_EXIT. For the value, enter the location (full path) of the authentication and authorization exit provided in the IVP migration files (*ivpauth.dll*). Click *Set* .
5. Select the DNA_AUTHOR_EXIT and enter the same location for that file (ivpauth.dll); then click *Set* .
6. Click *OK* to close the Client Properties dialog.
7. In the Management Console, select the server to be tested and, from the right-click menu, select *Properties* . The Server Properties dialog appears.
8. On the Database tab of the Server Properties dialog, specify the database name.
   Figure 4-2 Database tab of the Server Properties dialog



9. Click *OK* to close the Server Properties dialog.
10. In the Construction Workbench, select *Tools* > *Workbench Options* . Ensure that the settings on the Remote Prepare tab are correct. You can prepare to your local machine if you want so long as the local machine name is listed in the Machines list and the properties for it are correctly set.
11. In the Construction Workbench, open the component named IVP_C_SRV_C_CMP_DB_C. Follow the instructions in the comments in that file (to specify the name of the target database and the username and password used to access that target database).
12. Save the component to the AppBuilder repository.
13. If the server is not already running, start it.

Super Preparing the Server Partition

A server partition is an object that defines that a segment of an application is to be prepared on a server. There are three server partitions in IVP that can be Super Prepared: a C Server, an EJB or WebServices, and a Servlet.
To Super Prepare the server partition, complete the following steps:

1. Select the server partition to be Super Prepared.
2. Select **Build > Super Prepare** .
   All prepares should be successful.

C Server Tests

The following C server tests are discussed in this section:

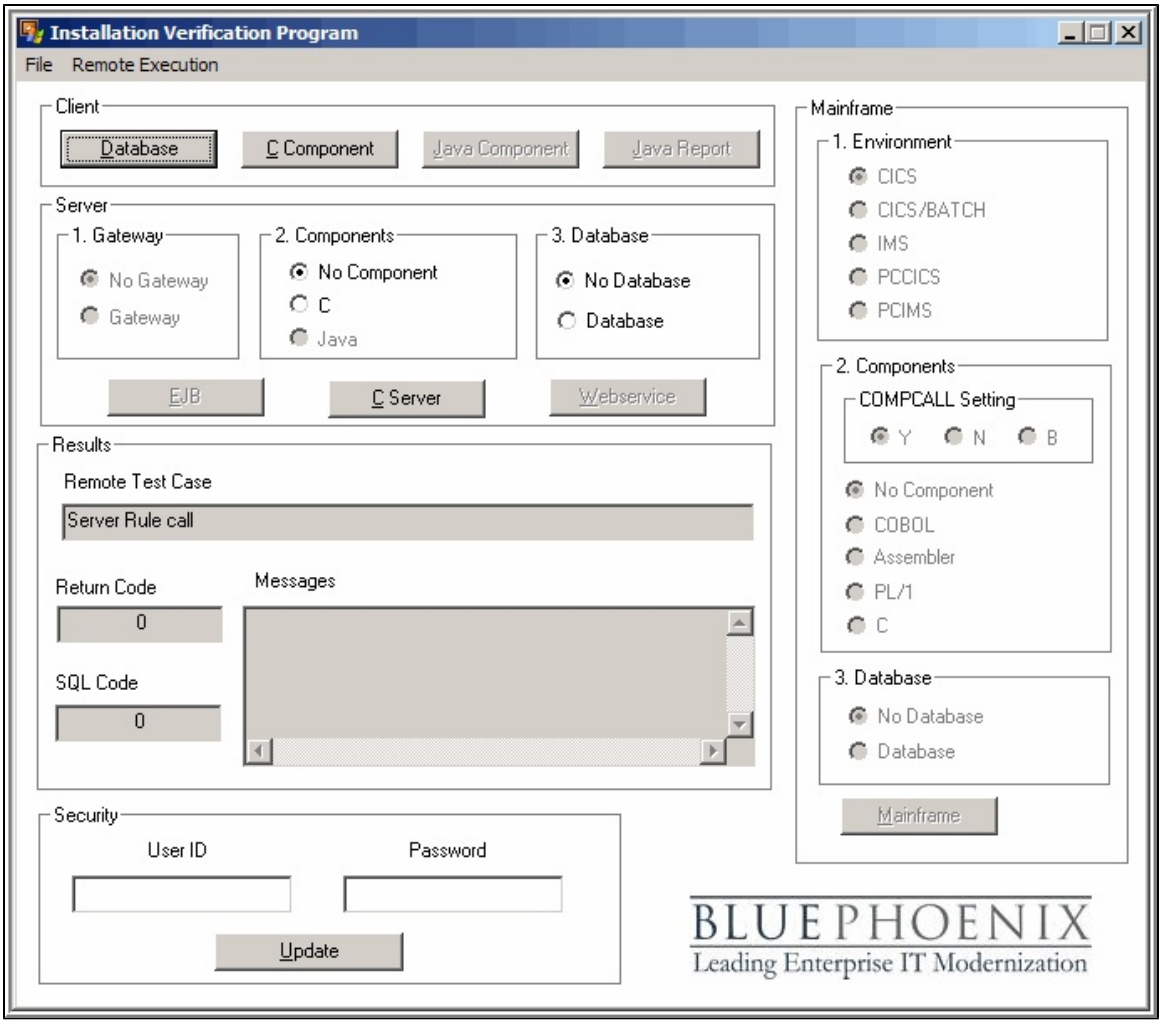
- C Server Rule Test
- C Server Database Rule Test
- C Server C Component Test
- C Server C Database Component Test

These tests assume that the C Server partition and the C Client partition have been configured and prepared successfully.

C Server Rule Test

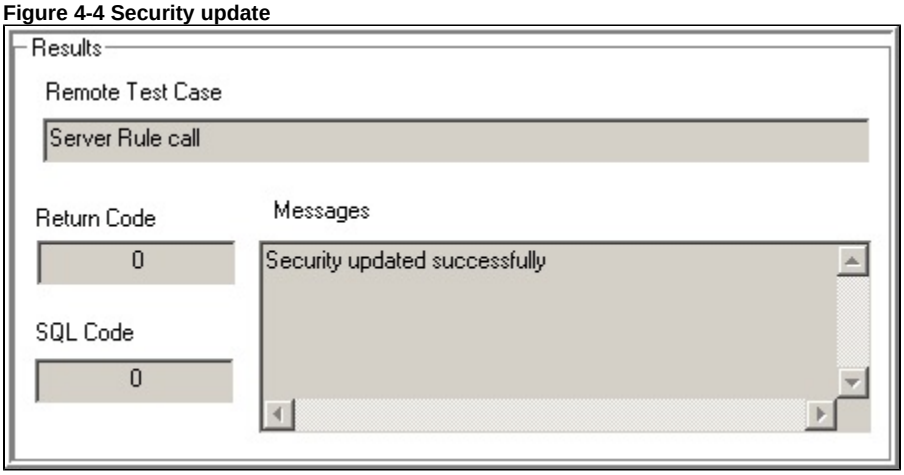
To test the C server rule:, complete the following steps:

1. Start the IVP application:
   a. In the Construction Workbench, select* ***Run > Windows***.*
   b. When the IVP Function menu appears, select **IVP Function > IVP Process** .
   c. Select whether to run the C Rule View (optional). The IVP command window appears.
   Figure 4-3 IVP Command Window



2. Enter your user ID and password in the Security section of the IVP Command window and click the **Update** button. The results section of the command window indicates that the update is successful.

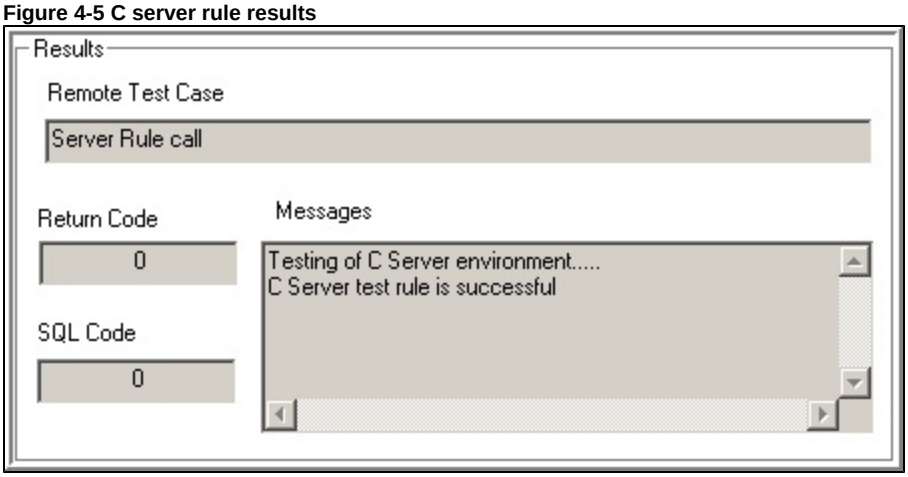
Figure 4-4 Security update



3. In the Server section of the IVP Command window click the *C Server* button. (Leave the Server section 2 Components selected with the **No Component** radio button. Leave the Server section 3 Database selected with the *No Database* radio button.) The results display in

the results field.

Figure 4-5 C server rule results



C Server Database Rule Test

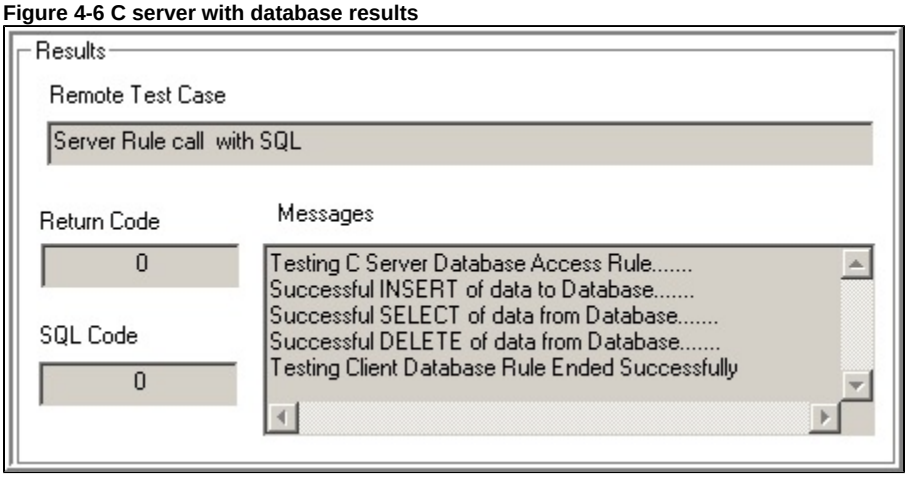
This section explains how to test a C Server database rule.

> **Note**
> The tester should have **insert, select,** and **delete** authority on the database table.

To test the database configuration, complete the following steps:

1. In the Server section of the IVP Command window leave the Server section 2 Components section with the*No Component* radio button selected.
2. In the Server section 3 Database select the **Database** radio button.
3. Click the **C Server** button. The results display in the results field.

Figure 4-6 C server with database results



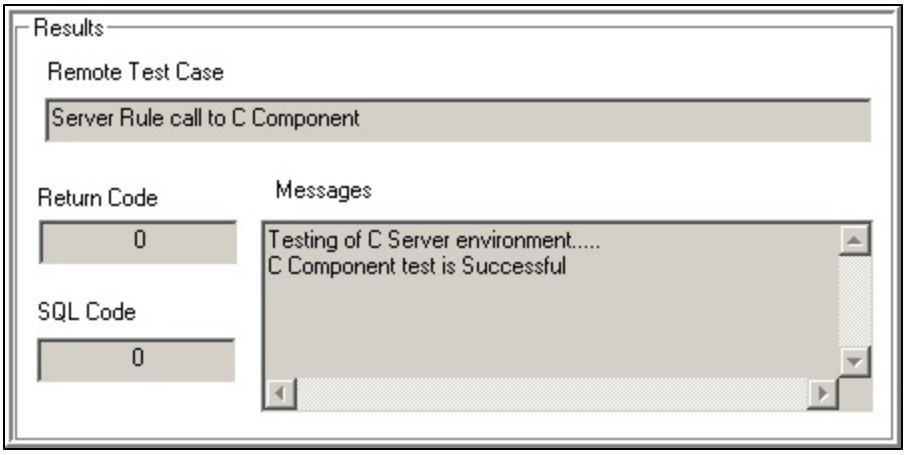
C Server C Component Test

To test the C Server calling a C User Component, complete the following steps:

1. In the Server section of the IVP Command window, in Server section 2 Components section select the* **C** *radio button.

1. In the Server section 3 Database select the *No Database* radio button.
2. Click the *C Server* button. The results display in the results field.

Figure 4-7 C server and C component results



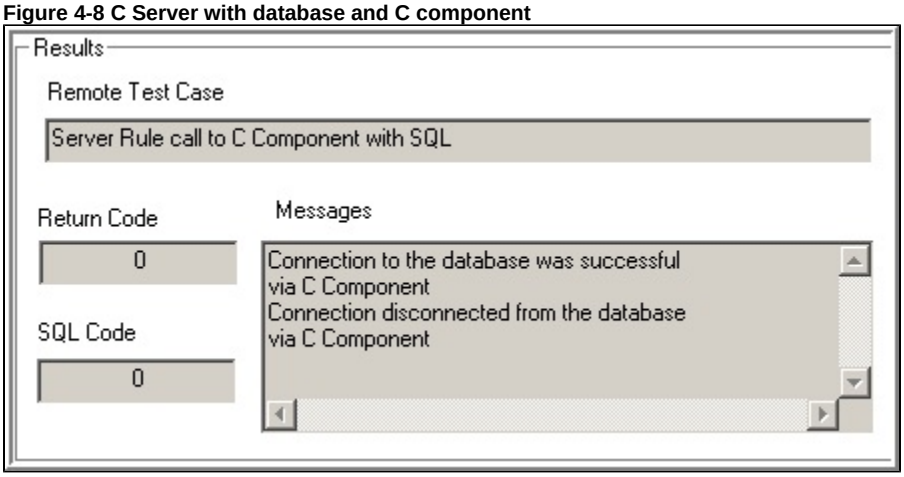
C Server C Database Component Test

> **Note**
> The tester should have **insert** , select,* and **delete** authority on the database table.

To test the C Server rule with a database and a C component, complete the following steps:

1. In the Server section of the IVP Command window, in the Server section 2 Components section select the* ***C*** *radio button.
2. In the Server section 3 Database select the **Database** radio button.
3. Click the **C Server** button. The results display in the results field.

Figure 4-8 C Server with database and C component



EJB and Webservices Tests

Configuration, preparation, and deployment of EJBs and WebServices is, in general, covered in *Deploying Applications Guide* .
The following EJB and Webservices tests are discussed in this section:

- EJB and Webservices Rule Test
- EJB and Webservices Database Rule Test
- Rule Test by way of a Gateway
- Database Rule by way of a Gateway
- EJB and Webservices C Component Test
- C Component Test by way of a Gateway
- EJB and Webservices C Database Component Test
- C Database Component Test by way of a Gateway
- Java Component Test
- Java Component Test by way of a Gateway
- Java Database Component Test
- Java Database Component Test by way of a Gateway

EJB and Webservices Rule Test

To perform an EJB or Webservices Rule test, complete the following steps on the IVP Command Window:

1. Select the* **No Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **No Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

EJB and Webservices Database Rule Test

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **No Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Rule Test by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **No Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Database Rule by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

EJB and Webservices C Component Test

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

#Select the **No Gateway** radio button.

1. Select the **C Component** radio button.
2. Select the **No Database** radio button.
3. Click **EJB** or **Webservices** , depending on the application server you are using.

C Component Test by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **Gateway** radio button.

1. Select the **C Component** radio button.
2. Select the **No Database** radio button.
3. Click **EJB** or **Webservices** , depending on the application server you are using.

EJB and Webservices C Database Component Test

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **No Gateway** radio button.
2. Select the **C Component** radio button.
3. Select the **Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

C Database Component Test by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **Gateway** radio button.
2. Select the **C Component** radio button.
3. Select the **Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Java Component Test

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **No Gateway** *radio button.
2. Select the **Java Component** radio button.
3. Select the **No Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Java Component Test by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **Gateway** radio button.
2. Select the *Java Component* radio button.
3. Select the *No Database* radio button.
4. Click *EJB* or *Webservices* , depending on the application server you are using.

Java Database Component Test

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **No Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **No Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Java Database Component Test by way of a Gateway

To perform an EJB or Webservices Rule test, complete the following steps on the Installation Verification Program Command Window:

1. Select the **No Gateway** radio button.
2. Select the **No Component** radio button.
3. Select the **No Database** radio button.
4. Click **EJB** or **Webservices** , depending on the application server you are using.

Servlet Tests

You can prepare, generate, and deploy a servlet to test the AppBuilder configuration with IVP.

Figure 4-9 Super Prepare of the Servlet Partition

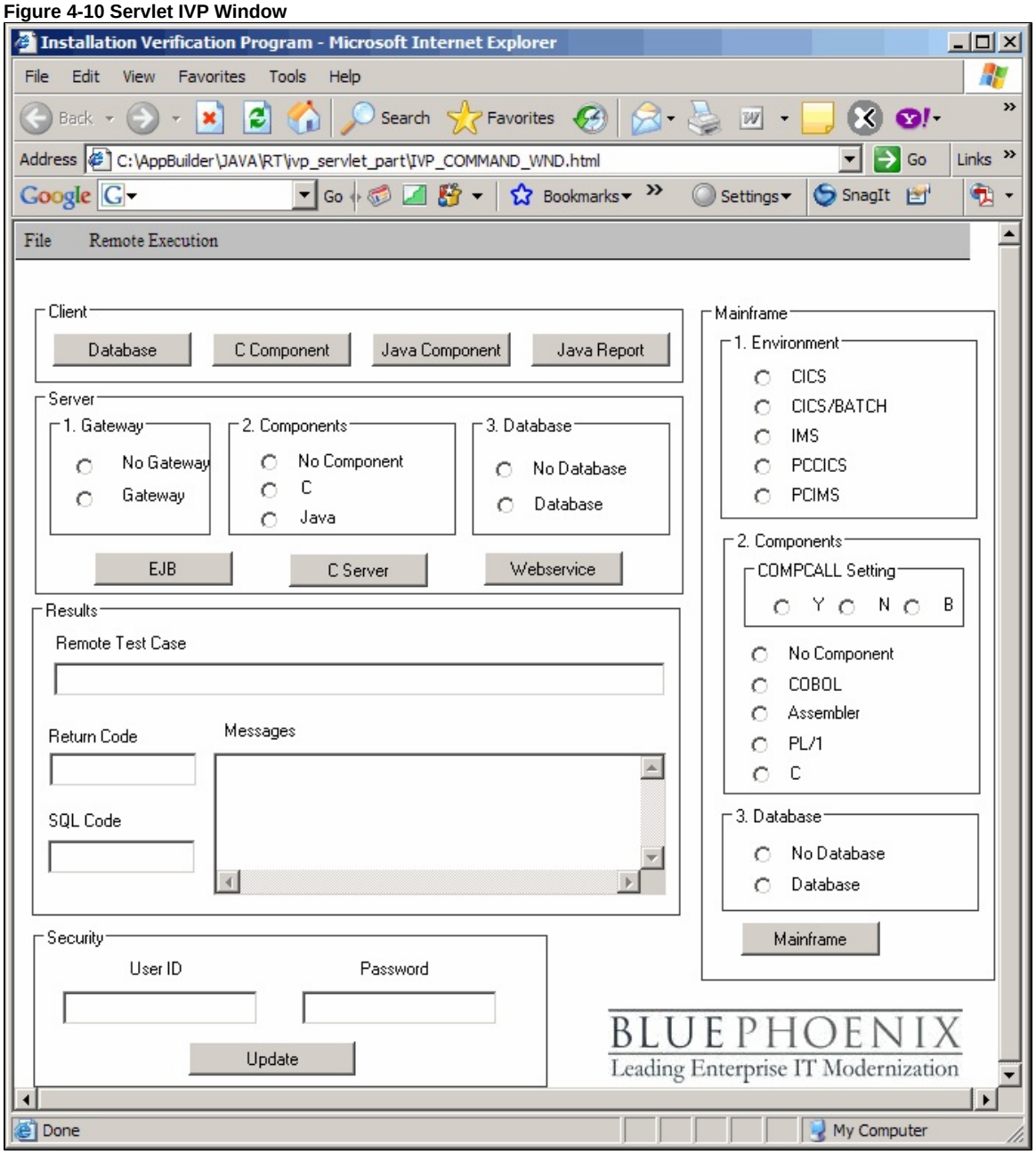
Name	Type	Submission St...	Submission Time	Last Modified	Platform
IVP_SERVLET_PART	PARTITION	Successful	11/29/2006 - 8:50:13 PM	11/27/2006 - 9:08:24 PM	Java
IVP_JAVA_RPT_EMP_FLE	FILE	Successful	11/29/2006 - 8:50:15 PM	11/27/2006 - 11:23:32 ...	Java
IVP_JAVA_RPT_DB	RULE	Successful	11/29/2006 - 8:50:17 PM	11/27/2006 - 11:23:32 ...	Java
IVP_BMP	BITMAP	Successful	11/29/2006 - 8:50:17 PM	11/27/2006 - 11:23:18 ...	Java
IVP_DB_FLE	FILE	Successful	11/29/2006 - 8:50:17 PM	11/27/2006 - 11:23:19 ...	Java
IVP_JAVA_CLT	COMPONENT	Successful	11/29/2006 - 8:50:17 PM	11/27/2006 - 11:23:18 ...	Java
IVP_COMMAND_WND	WINDOW	Successful	11/29/2006 - 8:50:18 PM	11/27/2006 - 11:23:17 ...	Java
IVP_SET_AUTHENTICATION	COMPONENT	Successful	11/29/2006 - 8:50:18 PM	11/27/2006 - 11:23:18 ...	Java
IVP_C_CLT	COMPONENT	Successful	11/29/2006 - 8:50:18 PM	11/27/2006 - 11:23:18 ...	Java
IVP_CLT_DB	RULE	Successful	11/29/2006 - 8:50:18 PM	11/27/2006 - 11:23:19 ...	Java
IVP_JAVA_RPT_RLE	RULE	Successful	11/29/2006 - 8:50:19 PM	11/27/2006 - 11:23:30 ...	Java
IVP_MVS_CLT_DRV	RULE	Successful	11/29/2006 - 8:50:19 PM	11/27/2006 - 11:23:23 ...	Java
IVP_DRV	RULE	Successful	11/29/2006 - 8:50:20 PM	11/27/2006 - 11:23:17 ...	Java
IVP_SERVLET_PART	PARTITION	Successful	11/29/2006 - 8:50:22 PM	11/27/2006 - 9:08:24 PM	Java
IVP_SERVLET_PART	PARTITION	Successful	11/29/2006 - 8:50:22 PM	11/27/2006 - 9:08:24 PM	Java

In the case of the servlet tests, you should be able to Super Prepare the Servlet partition. To deploy the war file that the servlet partition creates:

1. Copy the output war file from the <AppBuilder>\JAVA\RT\ivp_appcfg directory to the server.
2. Be sure also to copy the appbuilder.ini and appbuildercom.ini files to the server.
3. Make sure that the database has already been installed so that the application can access the data.
4. Unpack the war file.

5. Start the application by clicking on the command window. The servlet displays in an HTML (web) interface.

Figure 4-10 Servlet IVP Window



The servlet interface can be used for any or all of the tests available for the Java thick client, depending upon which tests have been prepared. Therefore, follow the procedures for the verification tests provided elsewhere in this book. See, for example:

- Verifying Java Runtime
- C Server Tests
- EJB and Webservices Tests

Mainframe Verifications

You can use the Installation Verification Program to verify a number of mainframe configurations. These tests check the ability to execute using IMS, PCIMS, CICS, CICS/BATCH, PCCICS and BATCH objects. You must first configure and prepare the IVP for the appropriate mainframe choices.

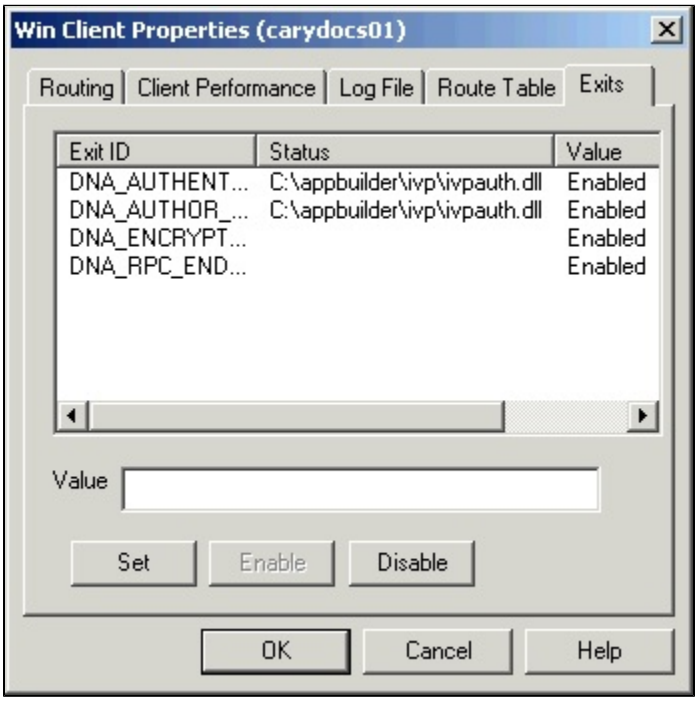
This chapter discusses tests for both AppBuilder 3.1n and AppBuilder 2035. Tests for AppBuilder 3.1n are prepared on the workstation and executed on the mainframe. AppBuilder 3.1n has no Enterprise Repository.

AppBuilder 2035 has an Enterprise Repository, and it requires a compiler and other files on the mainframe. Tests for AppBuilder 2035 can be prepared and executed on the mainframe, with the results sent back to the workstation.

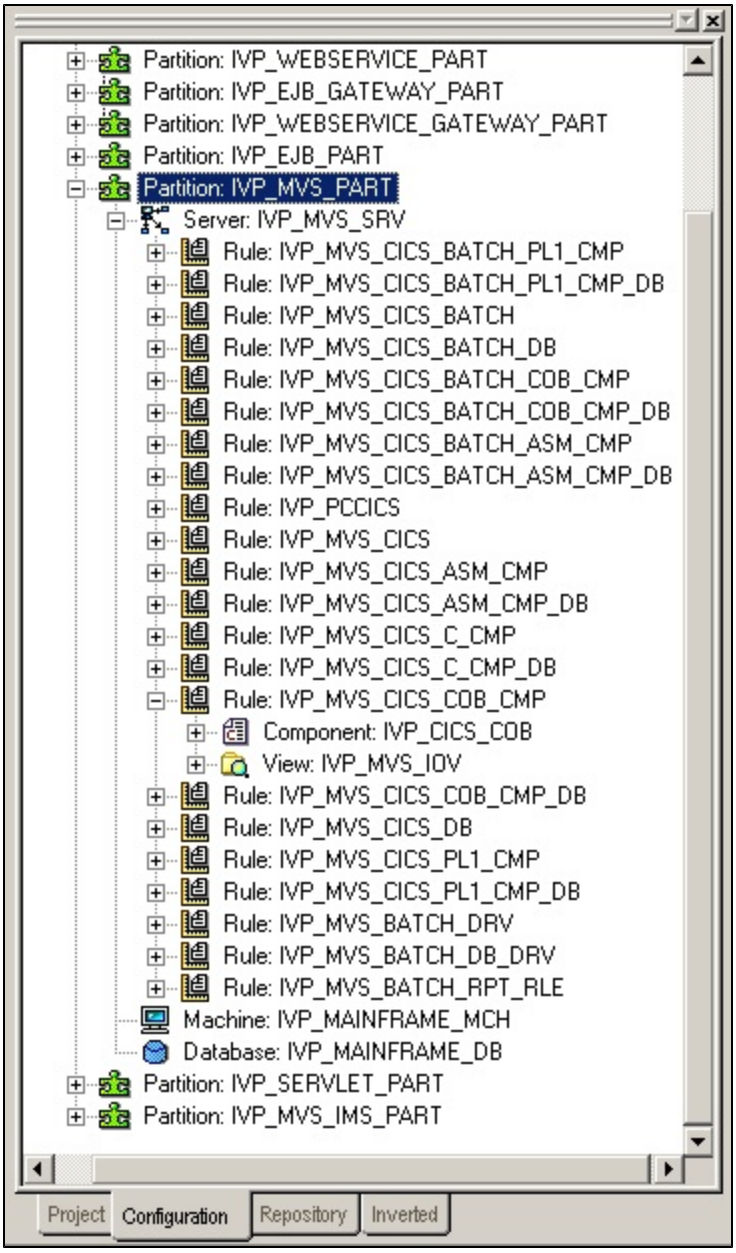
Configuring for Mainframe Testing

1. In order to run the IVP on the mainframe, authentication and authorization{*} exits must be specified for the client. To do this:
    a. Open the AppBuilder Management Console.
    b. Right-click the Windows client and select **Properties**. The Windows Client Properties dialog appears. (Figure 5-1 Exits tab)
    c. Select the Exits tab.
    d. Enter the location for the authentication and authorization exits. The *ivpauth.dll* file is provided in the IVP directory.
    e. Click the **Set** button to set the values and make sure that these exits are enabled.
    f. Click **OK** to close the Properties dialog.

Figure 5-1 Exits tab



2. In the Construction Workbench, select the IVP_MVS_PART partition.
Figure 5-2 IVP_MVS_PART partition



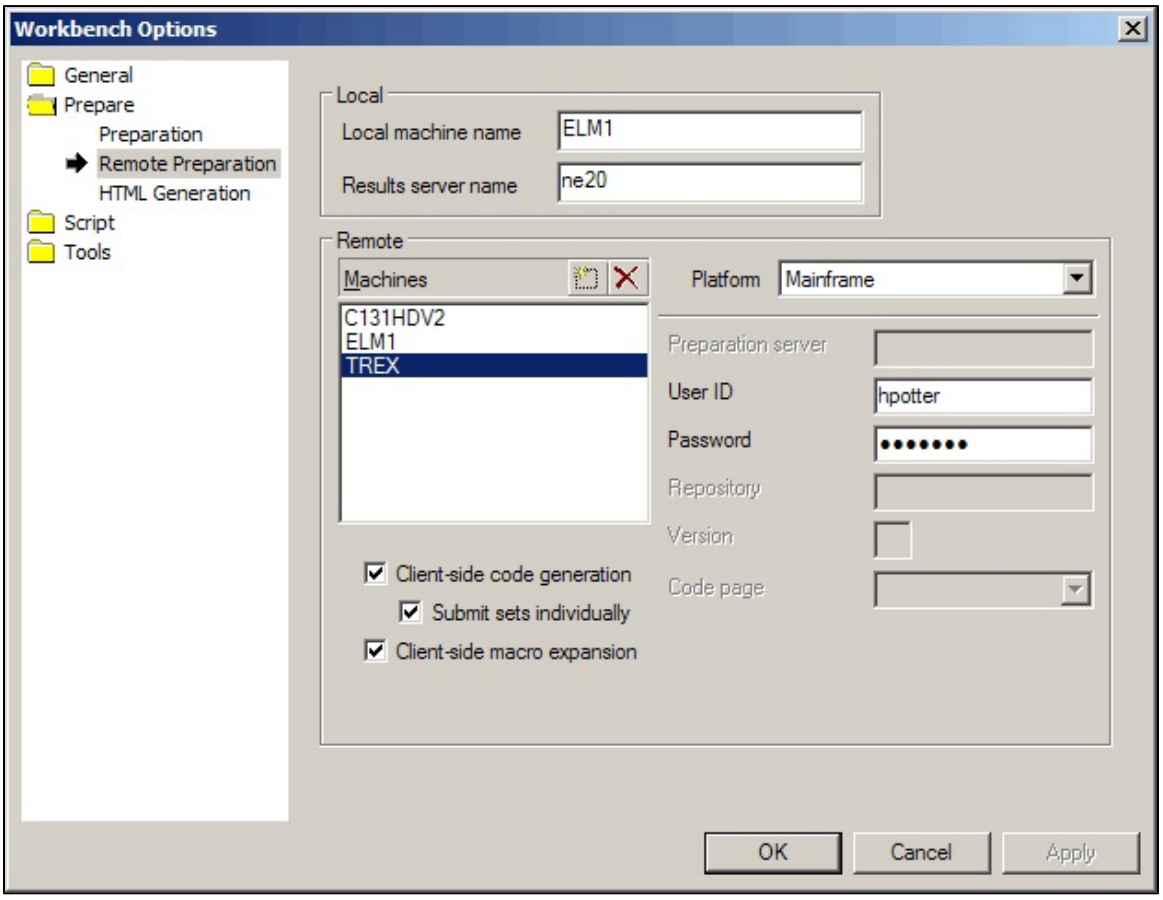
When you expand the partition, you see that the partition includes rules for CICS, CICS/BATCH, and BATCH.

> **Tip**
> If you attempt to Super Prepare the entire partition as a whole (by selecting the partition and selecting Super Prepare from the right-click menu), some rules will fail since you are attempting to prepare all the rules for the same environment. For this partition, do not Super Prepare the partition.

3. Select the Machine IVP_MAINFRAME_MCH (near the bottom of the partition), and, in the right-click menu, select *Properties* . The Properties dialog appears.
    a. Enter the appropriate Machine Group for the environment to be tested.
    b. Enter the Implementation Name [mainframe alias].
    c. Click the **Apply** button.
4. Select the Database IVP_MAINFRAME_DB (at the bottom of the partition), and, in the right-click menu, select **Properties** . The Properties dialog appears.
    a. Enter in the Database Directory Path the AppBuilder user qualifier for APPLDBRM.
    b. Enter the Implementation Name.
5. You must have the mainframe specified as a remote server for the Workbench. To add the mainframe as a remote server:
    a. In Construction Workbench, select **Tools > Workbench Options** and select the Remote Preparation tab. (**Figure 5-3 Remote Preparation tab of the Workbench Options dialog**)
    b. If the mainframe is not in the list of remote machines, add it.
    c. Specify the platform as Mainframe
    d. Enter your user id and password.

e. Click the **Apply** button.

Figure 5-3 Remote Preparation tab of the Workbench Options dialog



Super Preparing the MVS Mainframe Rules

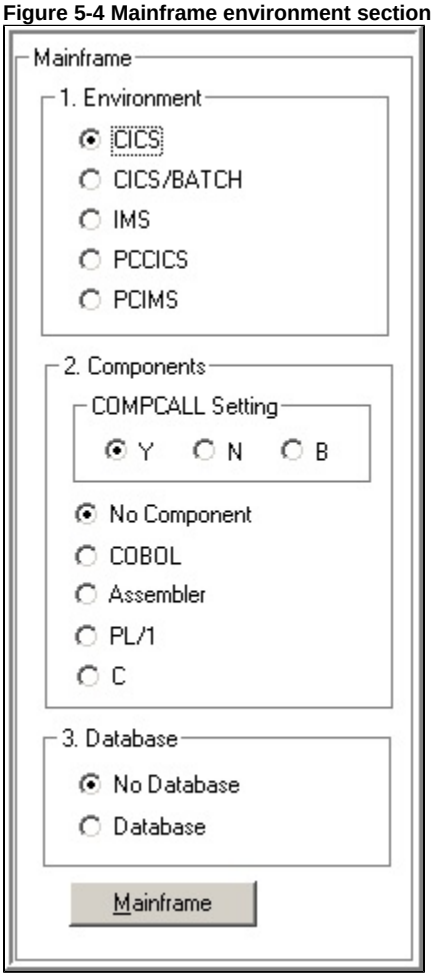
1. Select the rules for the environment to test.
   Table 5-1 Mainframe rules and environments in the MVS partition

Environment	Rules
Batch	IVP_MVS_BATCH_DRV IVP_MVS_BATCH_DB_DRV IVP_MVS_BATCH_RPT_RLE
CICS	IVP_MVS_CICS IVP_MVS_CICS_ASM_CMP IVP_MVS_CICS_ASM_CMP_DB IVP_MVS_CICS_C_CMP IVP_MVS_CICS_C_CMP_DB IVP_MVS_CICS_COB_CMP IVP_MVS_CICS_COB_CMP_DB IVP_MVS_CICS_DB IVP_MVS_CICS_PL1_CMP IVP_MVS_CICS_PL1_CMP_DB
PCCICS	IVP_PCCICS
CICS/BATCH	IVP_MVS_CICS_BATCH_PL1_CMP IVP_MVS_CICS_BATCH_PL1_CMP_DB IVP_MVS_CICS_BATCH IVP_MVS_CICS_BATCH_DB IVP_MVS_CICS_BATCH_COB_CMP IVP_MVS_CICS_BATCH_COB_CMP_DB IVP_MVS_CICS_BATCH_ASM_CMP IVP_MVS_CICS_BATCH_ASM_CMP_DB

2. Super prepare the rules selected for the environment to test. This preparation is done on the mainframe.

3. If you have not already done so, super prepare the client to be used (C or Java) as well.
4. Start the client (see Verifying C Runtime and Verifying Java Runtime) in order to open the IVP Command window.
5. In the IVP Command Window, on the Remote Execution menu select *Client/Mainframe* . Once this option has been selected the mainframe environment section of the IVP Command Window is enabled.

Figure 5-4 Mainframe environment section of the IVP Command Window



6. Specify which environment (Section 1) is to be tested and which test is to be run. The tests available from the MVS partition include:
Verifying the CICS Environment

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- C Component Test
- C Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test

Verifying the CICS/BATCH Environment

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test

Verifying the PCCICS Environment

- Mainframe Rule Test

Verifying the CICS Environment

In AppBuilder 3.1 the CICS/Batch tests have added a Comp Call setting dealing with component calls for verifying OpenCOBOL.

For the new Comp Call section of the Command Window:

Table 5-2 Comp Call Settings

Selection	Description
Y	ClassicCOBOL (Native)
N	View Information (Header)
B	Both

Mainframe Rule Test

To test the mainframe rule:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select **No Component** .
4. In the Database section, select **No Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.

> **Warning**
> The user ID and password should be all capital letters.

6. Click the **Mainframe** button. The client calls the mainframe to run the rule and reports the results.

Mainframe Database Rule Test

To test the mainframe rule with database:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select **No Component** .
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Component Test

To test the mainframe rule with an Assembler component:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **Assembler** radio button.
4. In the Database section, select **No Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Database Component Test

To test the mainframe rule with an Assembler component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **Assembler** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

C Component Test

To test the mainframe rule with a C component:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **C** radio button.
4. In the Database section, select **No Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

C Database Component Test

To test the mainframe rule with a C component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **C** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Component Test

To test the mainframe rule with a COBOL component:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **COBOL** radio button.
4. In the Database section, select **No Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Database Component Test

To test the mainframe rule with a COBOL component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **COBOL** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Component Test

To test the mainframe rule with a PL/1 component:

1. In the IVP Command Window, mainframe section, select **CICS** *environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **PL/1** radio button.
4. In the Database section, select **No Database**.
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Database Component Test

To test the mainframe rule with a PL/1 component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **PL/1** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Verifying the CICS/BATCH Environment

In AppBuilder 3.1 the CICS/Batch tests have added a Comp Call setting dealing with component calls for verifying OpenCOBOL calls.
For the new Comp Call section of the Command Window:
Table 5-3 Comp Call Settings

Selection	Description
Y	ClassicCOBOL (Native)
N	View Information (Header)
B	Both

Mainframe Rule Test

To test the mainframe rule:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select **No Component**.
4. In the Database section, select **No Database**.
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.

> **Warning**
> The user ID and password should be all capital letters.

6. Click the **Mainframe** button. The client calls the mainframe to run the rule and reports the results.

Mainframe Database Rule Test

To test the mainframe rule with database:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select **No Component**.
4. In the Database section, select **Database**.
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Component Test

To test the mainframe rule with an Assembler component:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **Assembler** radio button.
4. In the Database section, select **No Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Database Component Test

To test the mainframe rule with an Assembler component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **Assembler** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Component Test

To test the mainframe rule with a COBOL component:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **COBOL** radio button.
4. In the Database section, select **No Database**.
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Database Component Test

To test the mainframe rule with a COBOL component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **COBOL** radio button.
4. In the Database section, select **Database** .

5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Component Test

To test the mainframe rule with a PL/1 component:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **PL/1** radio button.
4. In the Database section, select **No Database**.
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Database Component Test

To test the mainframe rule with a PL/1 component calling a database:

1. In the IVP Command Window, mainframe section, select **CICS/Batch** environment.
2. In the Component section, select the appropriate Comp Call setting.
3. In the Component section, select the **PL/1** radio button.
4. In the Database section, select **Database** .
5. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
6. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Verifying the PCCICS Environment

The PCCICS verifications include

- Mainframe Rule Test.

Mainframe Rule Test

To test the mainframe rule:

1. In the IVP Command Window, mainframe section, select **PCCICS** environment.
2. In the Component section, select **No Component** .
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.

   Warning
   The user ID and password should be all capital letters.

5. Click the *Mainframe* button. The client calls the mainframe to run the rule and reports the results.

Verifying the Batch Environment

You can remote prepare the Batch objects, but you can't execute them on the workstation. The Batch rules must be executed on the mainframe. Assuming that the Batch rules prepared successfully, the following covers the tests in this environment:

ClassicCOBOL Tests

Follow the steps:

1. Execute JCL member IVPBATCH
2. Rule test:
   The following should appear in the SYSOUT of the job:
   Return Code 00000
   Status: Batch Rule Execution Pass
3. Assembler Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch Assembler Component Pass
4. COBOL Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch COBOL Component Pass

5. C Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch C Component Pass
6. PL/1 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch PLI Component Pass

ClassicCOBOL DB2 Tests

Follow the steps:

1. Execute JCL member IVPBTDB2
2. DB2 Rule test:
   The following should appear in the SYSOUT of the job:
   Return Code 00000
   SQL Code: 000000
   Status: Batch DB2 Rule Pass
3. Assembler DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch Assembler DB2 Component Pass
4. COBOL DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch DB2 COBOL Component Pass
5. C DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch C DB2 Component Pass
6. PL/1 DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch PLI DB2 Component Pass

OpenCOBOL Tests

Follow the steps:

1. Execute JCL member IVPOCBT
2. Rule test:
   The following should appear in the SYSOUT of the job:
   Return Code 00000
   Status: Batch Rule Execution Pass
3. Assembler Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch Assembler Component Pass
4. COBOL Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch COBOL Component Pass
5. C Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch C Component Pass
6. PL/1 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   Batch PLI Component Pass

OpenCOBOL DB2 Tests

Follow the steps:

1. Execute JCL member IVPOCDB*
2. DB2 Rule test:
   The following should appear in the SYSOUT of the job:

Return Code 00000
SQL Code: 000000
Status: Batch DB2 Rule Pass

3. Assembler DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch Assembler DB2 Component Pass
4. COBOL DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch DB2 COBOL Component Pass
5. C DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch C DB2 Component Pass
6. PL/1 DB2 Component Test:
   The following should appear in the IVPOUT of the job:
   Return Code: 4
   SQL Code: 0
   Batch PLI DB2 Component Pass

Batch Report Test

Follow the steps:

1. Execute JCL member IVPRPT.*
2. Generate a mainframe report test.

Preparing the IMS Mainframe Rules

You cannot Super Prepare the IVP_MVS_IMS_PART partition on the Client. This partition must be prepared on the mainframe.

Verifying the IMS Environment

The IMS tests include:

- Mainframe Rule Test
- Mainframe Database Rule Test
- Assembler Component Test
- Assembler Database Component Test
- COBOL Component Test
- COBOL Database Component Test
- PL/I Component Test
- PL/I Database Component Test
- C Component Test
- C Database Component Test

Mainframe Rule Test

To test the mainframe rule:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select **No Component**.
3. In the Database section, select **No Database**.
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.

> **Warning**
> The user ID and password should be all capital letters.

5. Click the **Mainframe** button. The client calls the mainframe to run the rule and reports the results.

Mainframe Database Rule Test

To test the mainframe rule with database:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select **No Component** .
3. In the Database section, select **Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Component Test

To test the mainframe rule with an Assembler component:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **Assembler** radio button.
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Assembler Database Component Test

To test the mainframe rule with an Assembler component calling a database:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **Assembler** radio button.
3. In the Database section, select **Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Component Test

To test the mainframe rule with a COBOL component:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **COBOL** radio button.
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

COBOL Database Component Test

To test the mainframe rule with a COBOL component calling a database:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **COBOL** radio button.
3. In the Database section, select **Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Component Test

To test the mainframe rule with a PL/1 component:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **PL/1** radio button.
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

PL/I Database Component Test

To test the mainframe rule with a PL/1 component calling a database:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **PL/1** radio button.
3. In the Database section, select **Database**.
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

C Component Test

To test the mainframe rule with a C component:

1. In the IVP Command Window, mainframe section, select **IMS** environment.
2. In the Component section, select the **C** radio button.
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

C Database Component Test

To test the mainframe rule with a C component calling a database:

1. In the IVP Command Window, mainframe section, select **IMS** environment.

1. In the Component section, select the **C** radio button.
2. In the Database section, select **Database** .
3. Enter your mainframe user ID and password in the Security section of the IVP Command Window.
4. Click the **Mainframe** button. The client calls the mainframe to run the rule and access the database and then reports the results.

Verifying the PCIMS Environment

The PCIMS environment includes the

- Mainframe Rule Test.

Mainframe Rule Test

To test the mainframe rule:

1. In the IVP Command Window, mainframe section, select **PCIMS** environment.
2. In the Component section, select **No Component** .
3. In the Database section, select **No Database** .
4. Enter your mainframe user ID and password in the Security section of the IVP Command Window.

> **Warning**
> The user ID and password should be all capital letters.

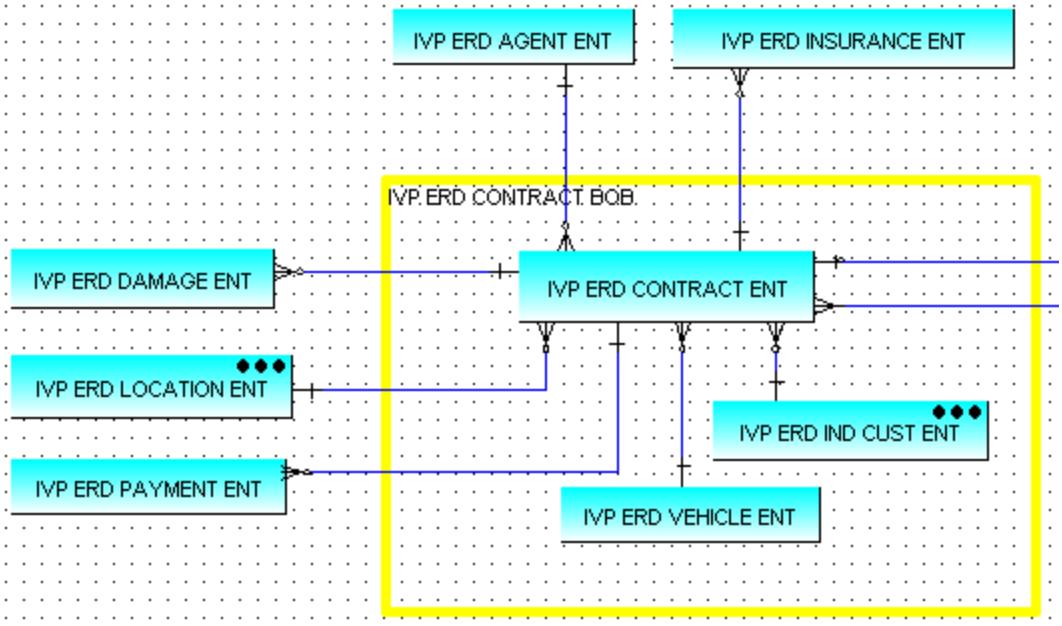
5. Click the **Mainframe** button. The client calls the mainframe to run the rule and reports the results.

Other IVP Features

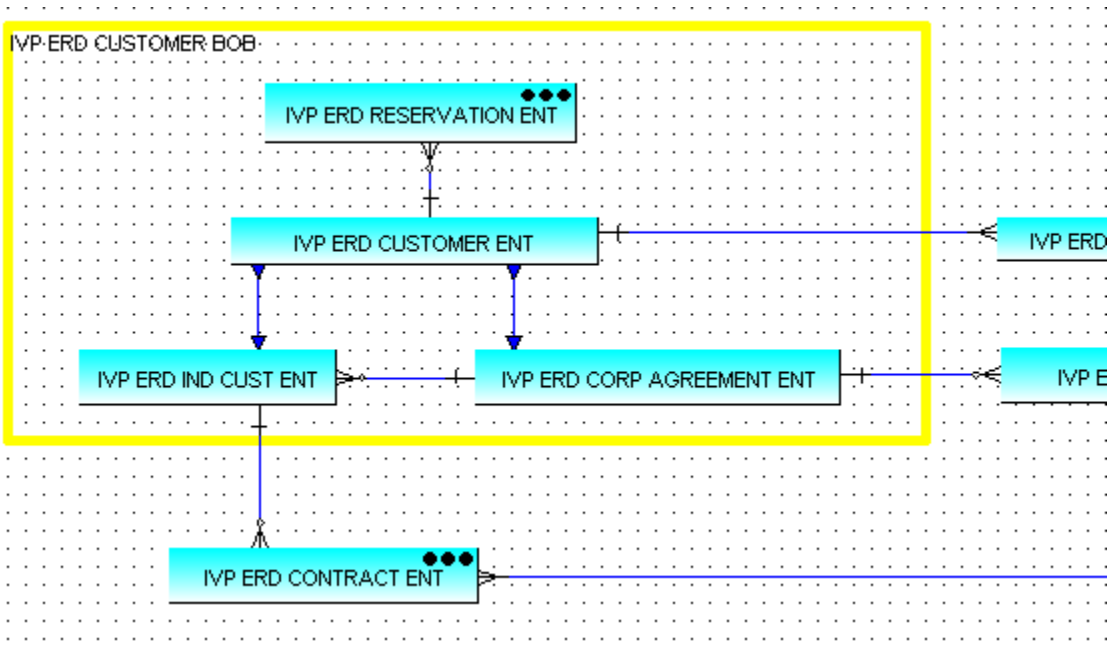
When you import and install IVP into an AppBuilder repository, you have the IVP application written in AppBuilder's Rules Language to prepare, build, and use for testing whatever environment you use. Those tests are covered in the previous sections.
In addition, however, the IVP provides a number of objects which may prove to be useful as illustrations. These include:

IVP Objects			
Object	**Type**	**Name**	**Description**
Entity Relationship Diagrams		IVP_ERD_EX_CONTRACT IVP_ERD_EX_CUSTOMER IVP_ERD_EX_RESERVATION	
Report		IVP_JAVA_RPT IVP_MVS_RPT	See Setting Up the Database Tables and Java Reports with Java Runtime.
Window Flow Diagram		IVP_WFD	

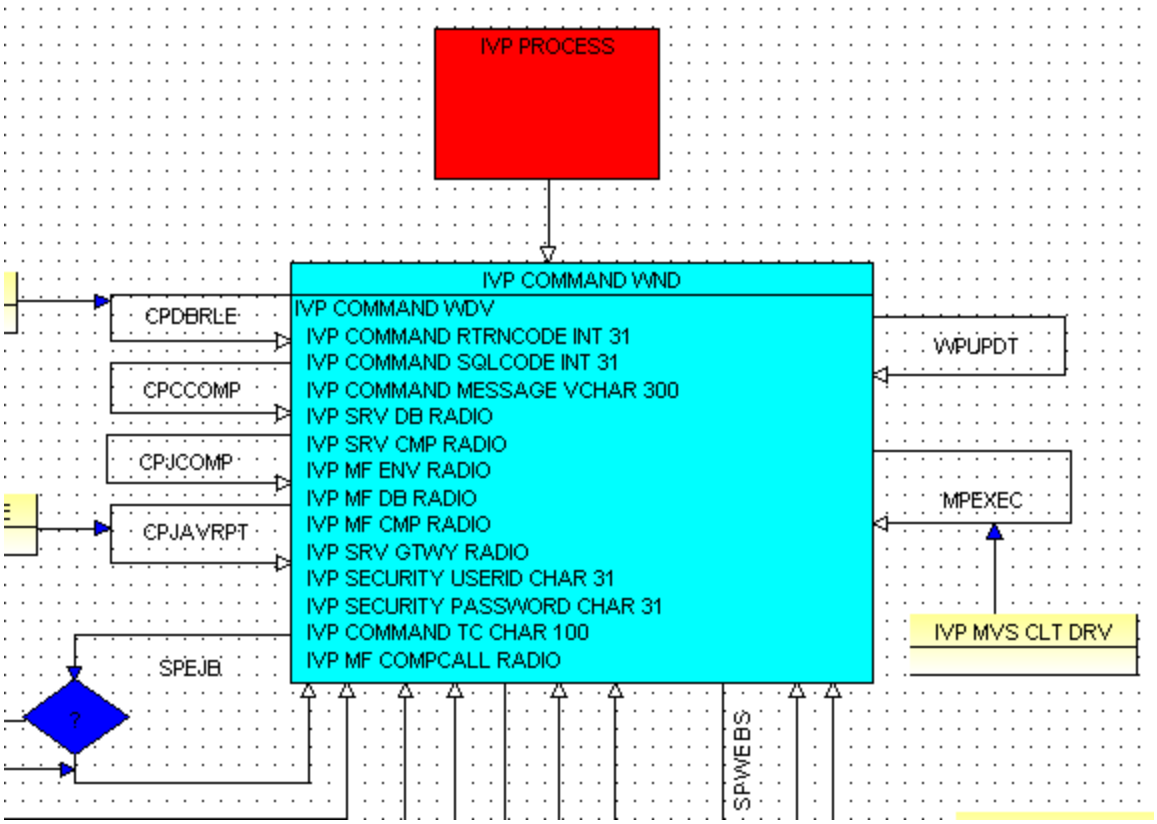
Entity Relationship Diagram example: IVP_ERD_EX_CONTRACT



Entity Relationship Diagram example: IVP_ERD_EX_CUSTOMER



Window Flow Diagram example


IVP PROCESS
IVP COMMAND WND
IVP COMMAND WDV
IVP COMMAND RTRNCODE INT 31
IVP COMMAND SQLCODE INT 31
IVP COMMAND MESSAGE VCHAR 300
IVP SRV DB RADIO
IVP SRV CMP RADIO
IVP MF ENV RADIO
IVP MF DB RADIO
IVP MF CMP RADIO
IVP SRV GTWY RADIO
IVP SECURITY USERID CHAR 31
IVP SECURITY PASSWORD CHAR 31
IVP COMMAND TC CHAR 100
IVP MF COMPCALL RADIO
CPDBRLE
CPCCOMP
CPJCOMP
CPJAVRPT
SPEJB
?
WPUPDT
MPEXEC
IVP MVS CLT DRV
SPWEBS