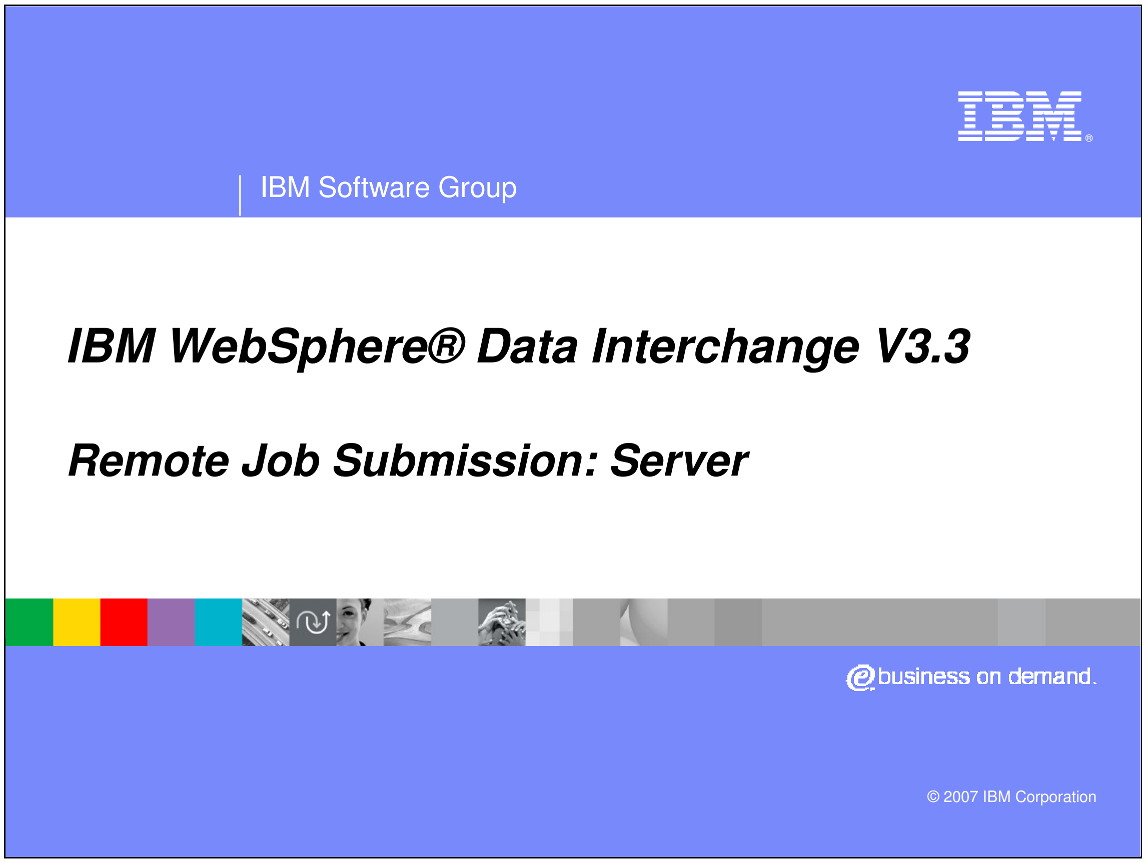
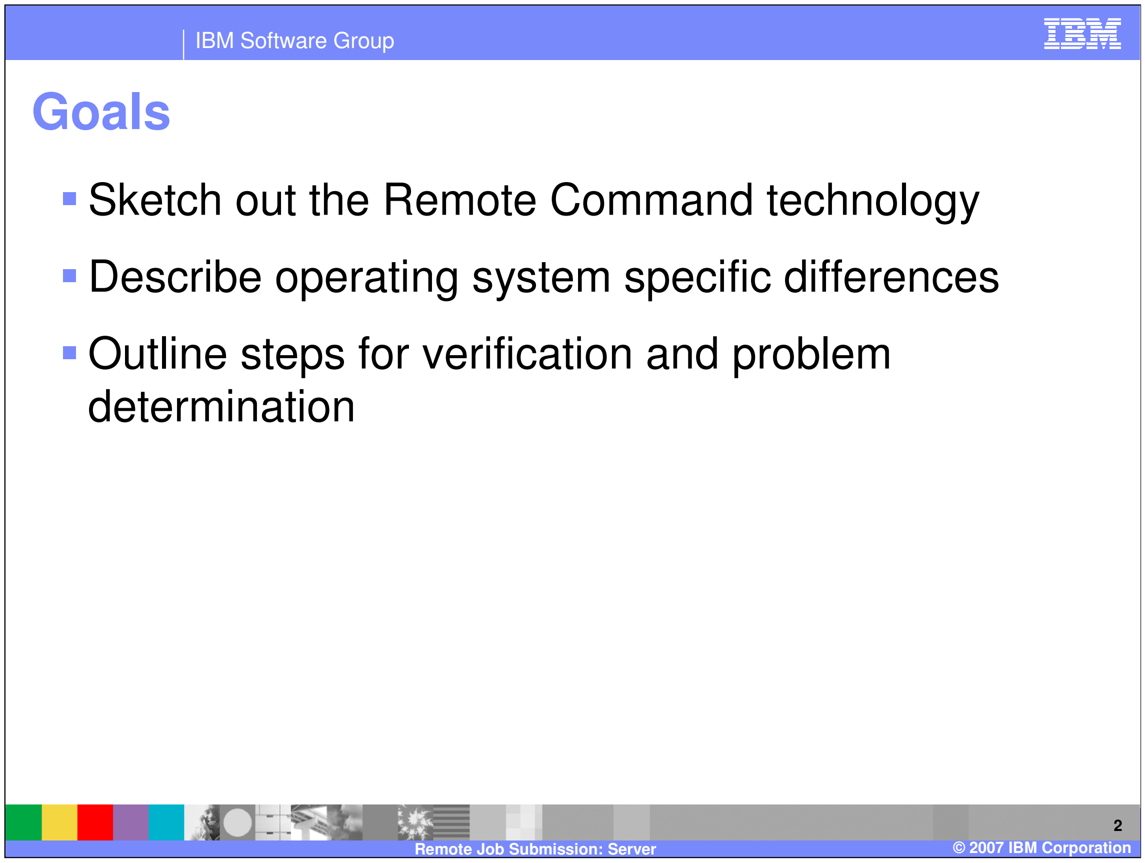

IBM Software Group

# IBM WebSphere® Data Interchange V3.3

## Remote Job Submission: Server

business on demand.

This presentation covers the implementation of Remote Job Submission for WebSphere Data Interchange version 3.3. This includes discussion of the code and technology behind the Client features that have been added with this release.



IBM Software Group

## Goals

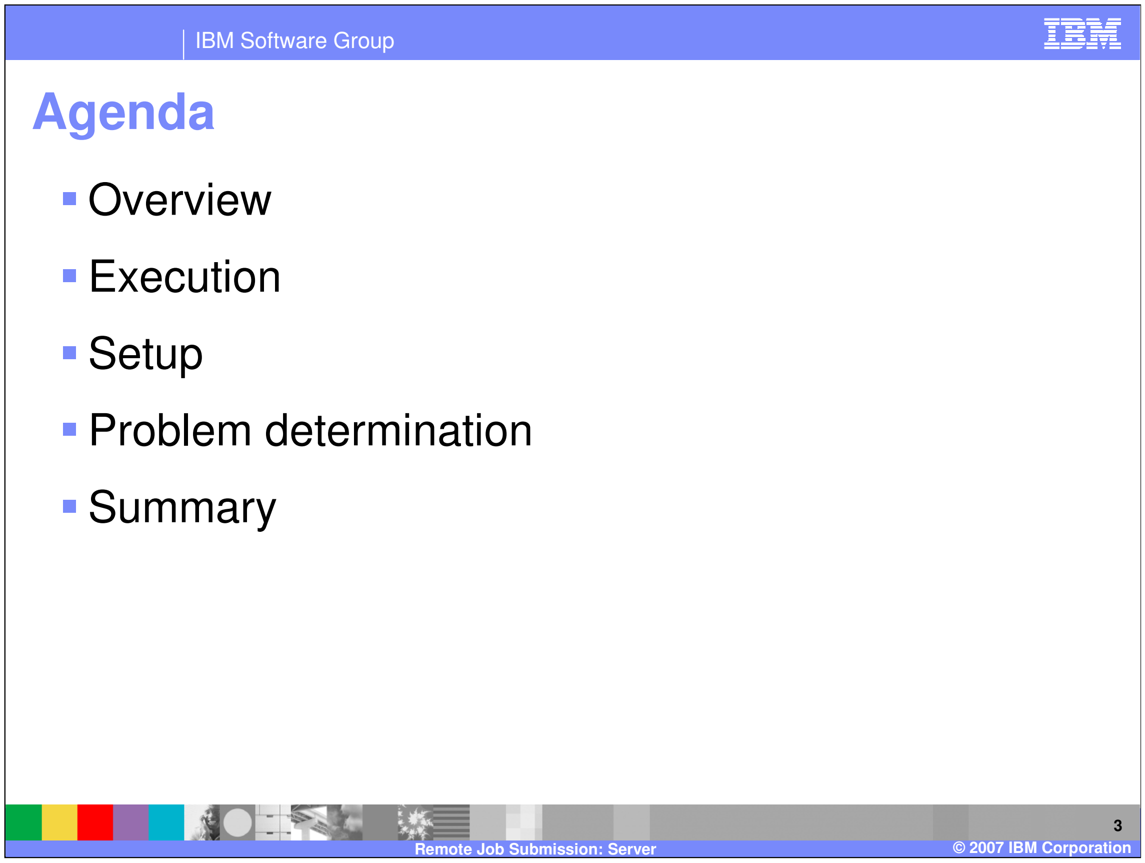
- Sketch out the Remote Command technology
- Describe operating system specific differences
- Outline steps for verification and problem determination

2

Remote Job Submission: Server

© 2007 IBM Corporation

This presentation strives to remove the hesitation a Client user might have when using features in the Submissions or Document Store functional areas. It overviews the solution in an easily understood context, gives guidance on operating system idiosyncrasies that might otherwise be frustrating, and points out areas of failure and techniques for identifying specific problems.



IBM Software Group

## Agenda

- Overview
- Execution
- Setup
- Problem determination
- Summary

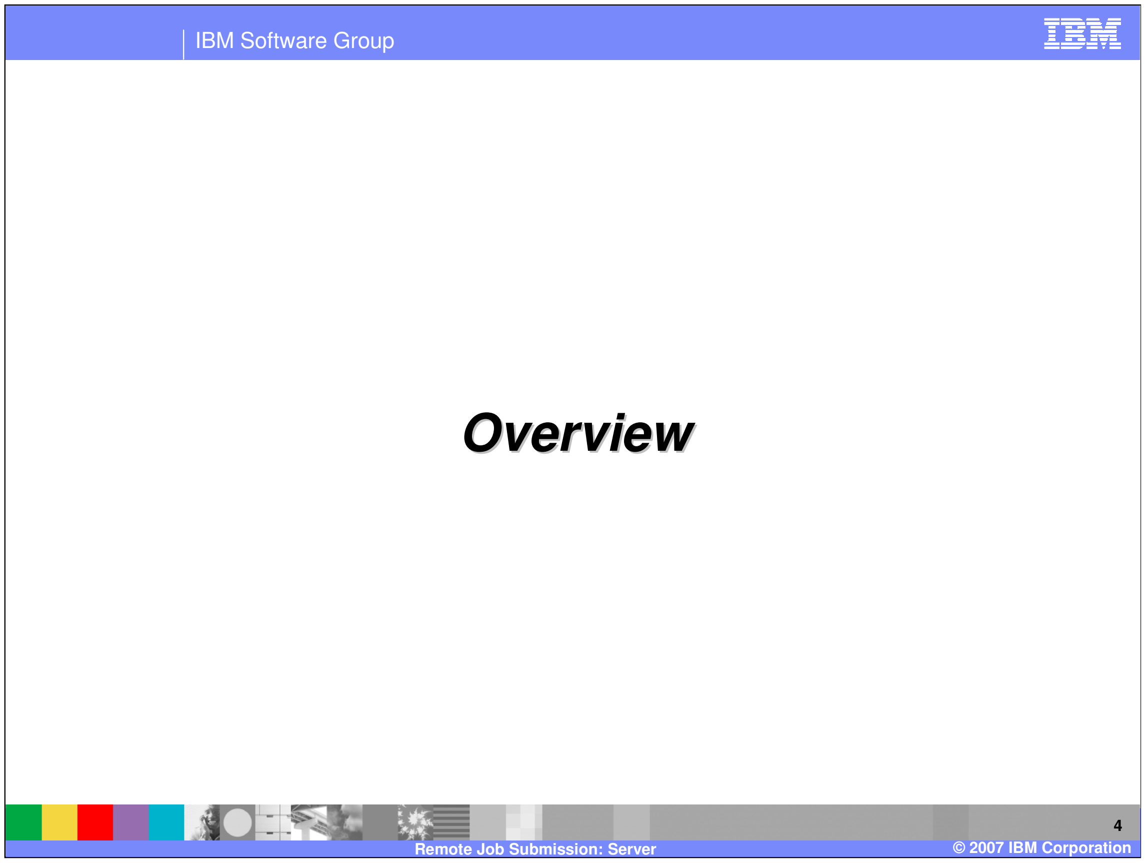
3

Remote Job Submission: Server

© 2007 IBM Corporation

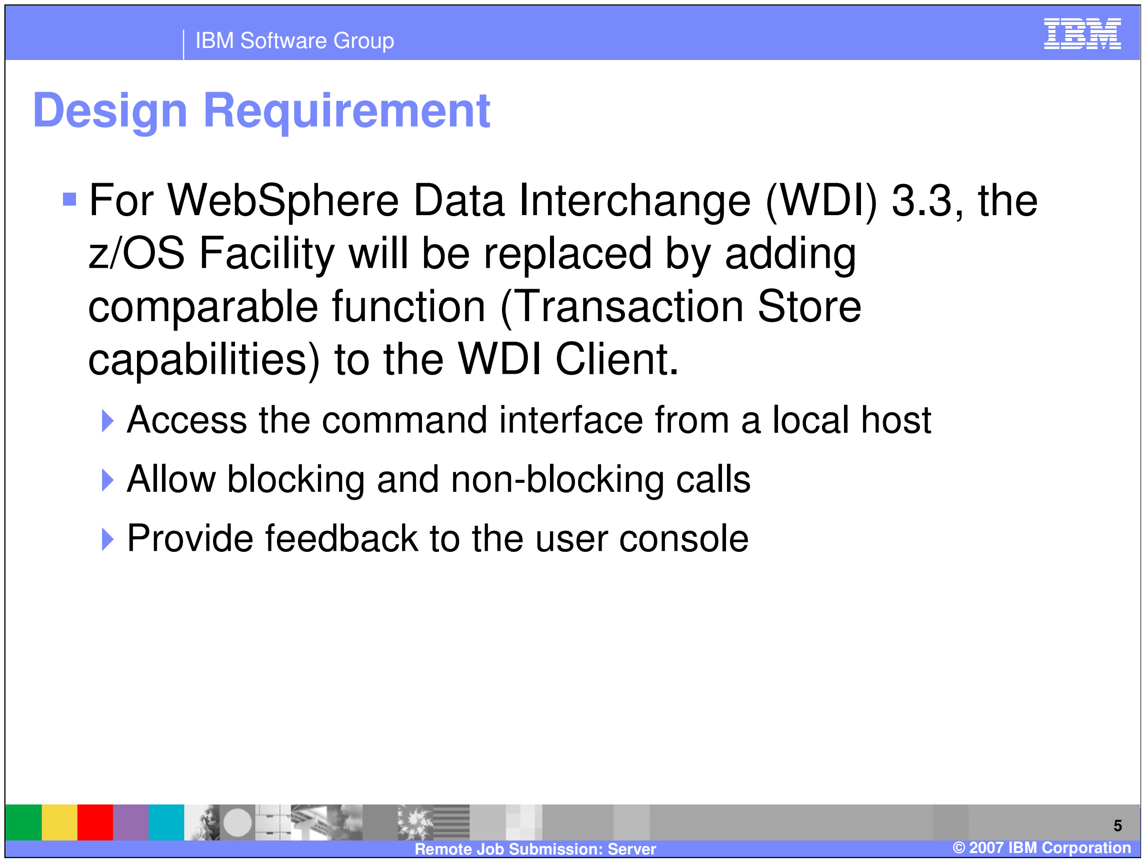
The content separates into four areas: an overview of the solution, a description of typical execution, information about infrastructure setup, and notes on problem determination.

These sections are followed by a summary to reinforce these concepts.



## Overview

This section covers the impetus for this change and the components of the solution.



IBM Software Group

## Design Requirement

- For WebSphere Data Interchange (WDI) 3.3, the z/OS Facility will be replaced by adding comparable function (Transaction Store capabilities) to the WDI Client.
  - Access the command interface from a local host
  - Allow blocking and non-blocking calls
  - Provide feedback to the user console

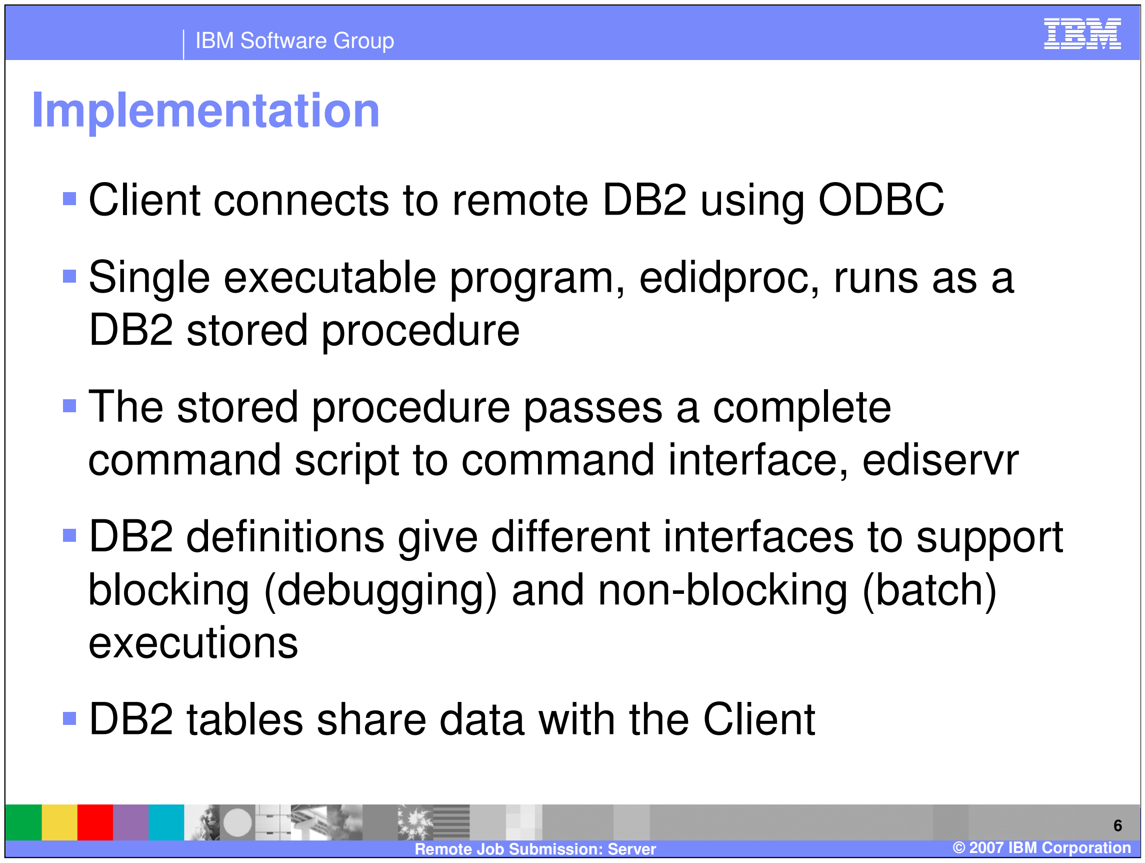
5

Remote Job Submission: Server

© 2007 IBM Corporation

Remote Job Submission stems from a need to expand WebSphere Data Interchange (WDI) Client features to include functions that formerly were available only through the z/OS interface. Though eclipsed by current technology, the original Character User Interface (CUI) offered powerful features. These included the ability to enter and process any PERFORM command understood by the WDI command interpreter, the ability to free the terminal so that a command ran in background, and the ability to get feedback on background processes.

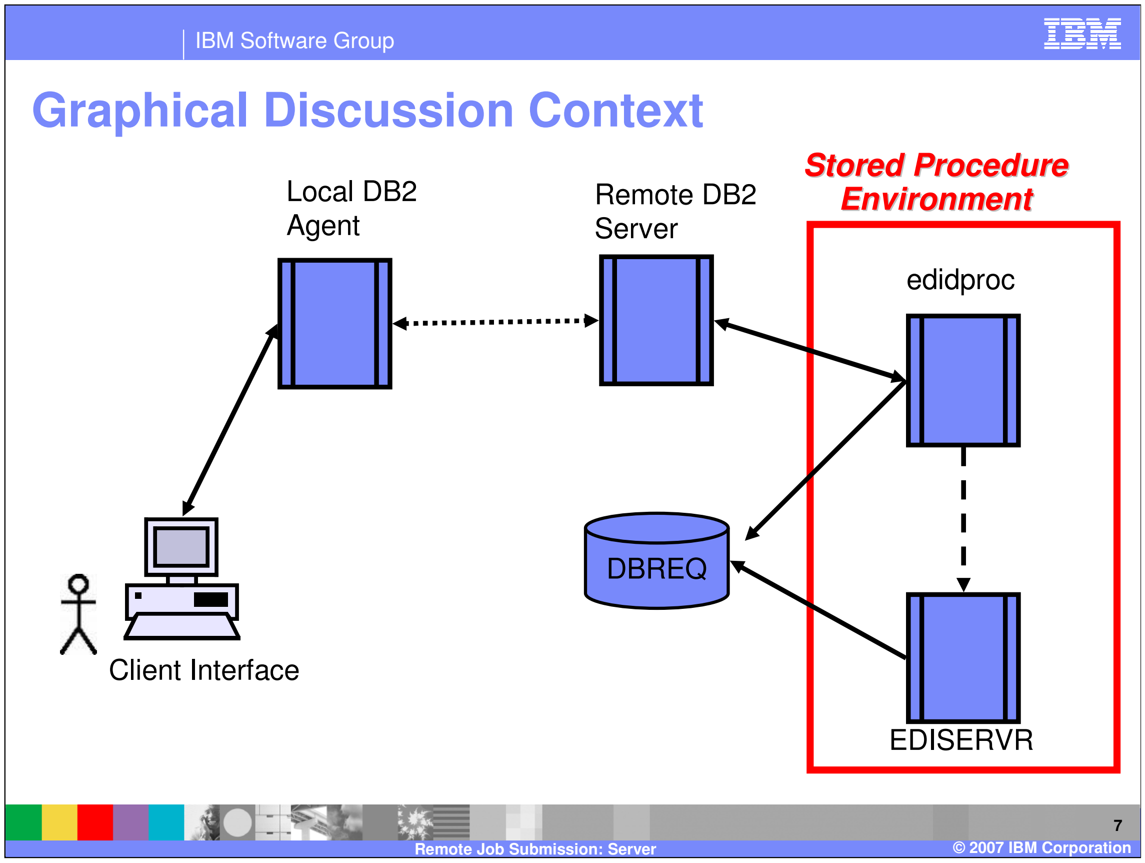
These powerful features were available only to z/OS clients running the CUI on a local database. This change also included the mandate to make them available on all platforms through the Windows-based GUI.



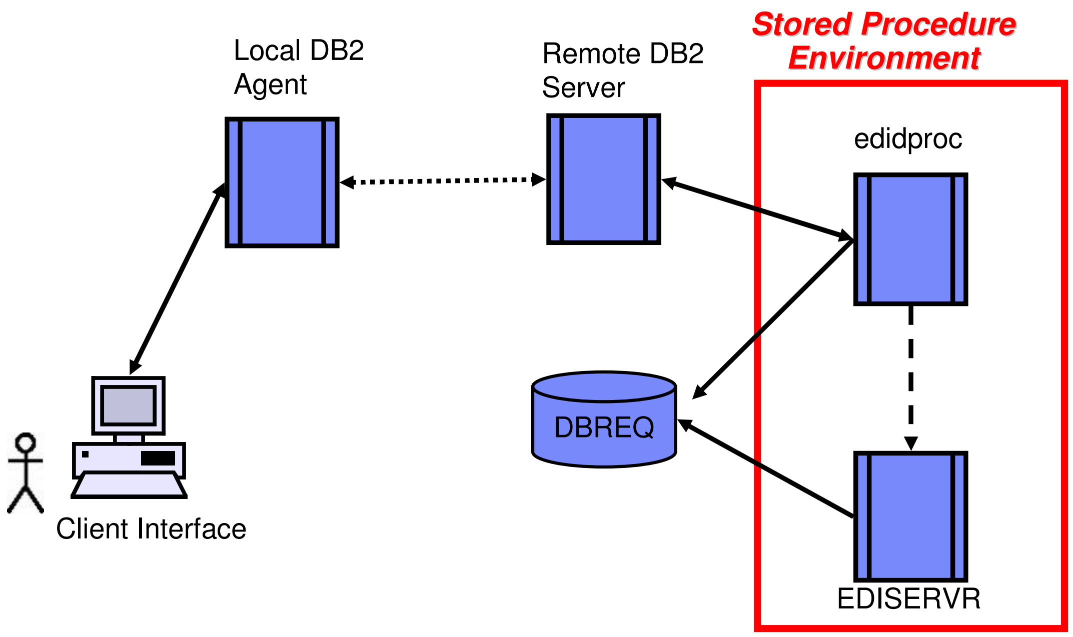
## Implementation

- Client connects to remote DB2 using ODBC
- Single executable program, edidproc, runs as a DB2 stored procedure
- The stored procedure passes a complete command script to command interface, ediservr
- DB2 definitions give different interfaces to support blocking (debugging) and non-blocking (batch) executions
- DB2 tables share data with the Client

The architectural solution is to use DB2 Stored Procedures. Since WDI Client already connects to each database node using ODBC, so a stored procedure operating in that same node can access the data that is being presented. This change ported the command script interface to z/OS so that the Client could construct and send an identical script without having to customize its behavior for each platform. The program interface accepts a blocking indicator so that tasks can operate in the background similar to the original host panel interface. Since output files would be local to the execution node, the server I/O services expanded to use DB2 entities accessible at the remote node and to the local Client through ODBC.


IBM Software Group

## Graphical Discussion Context



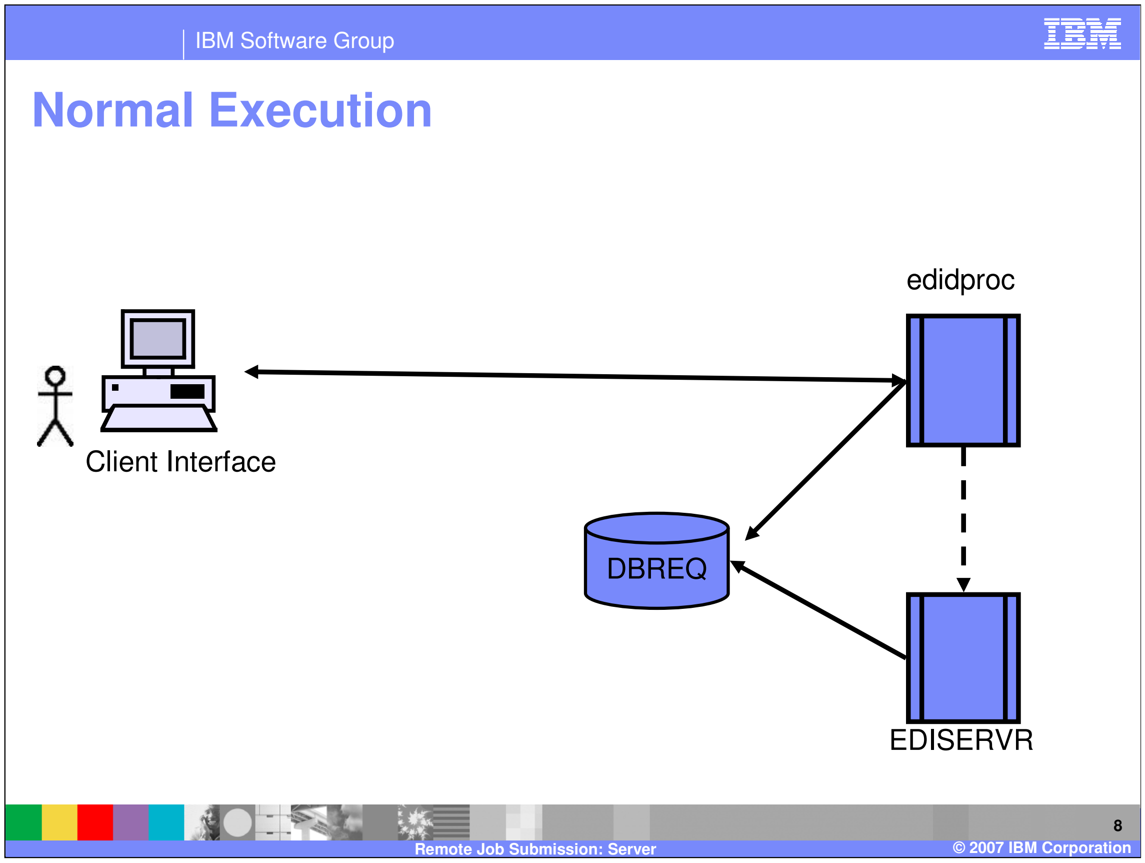
7

Remote Job Submission: Server

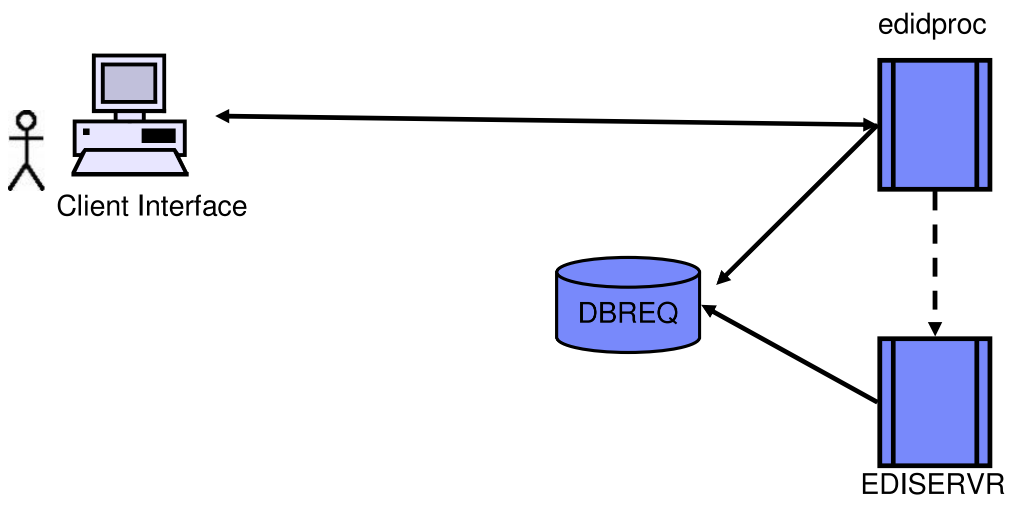
© 2007 IBM Corporation

This diagram gives a graphical depiction of the basic components involved in remote execution. The local DB2 agent and the remote DB2 server are critical to this design; they provide the ability to decouple the interface from the remote host. Since this is already a part of the WDI solution, it does not add additional complexity to this design and does not need to be described here.

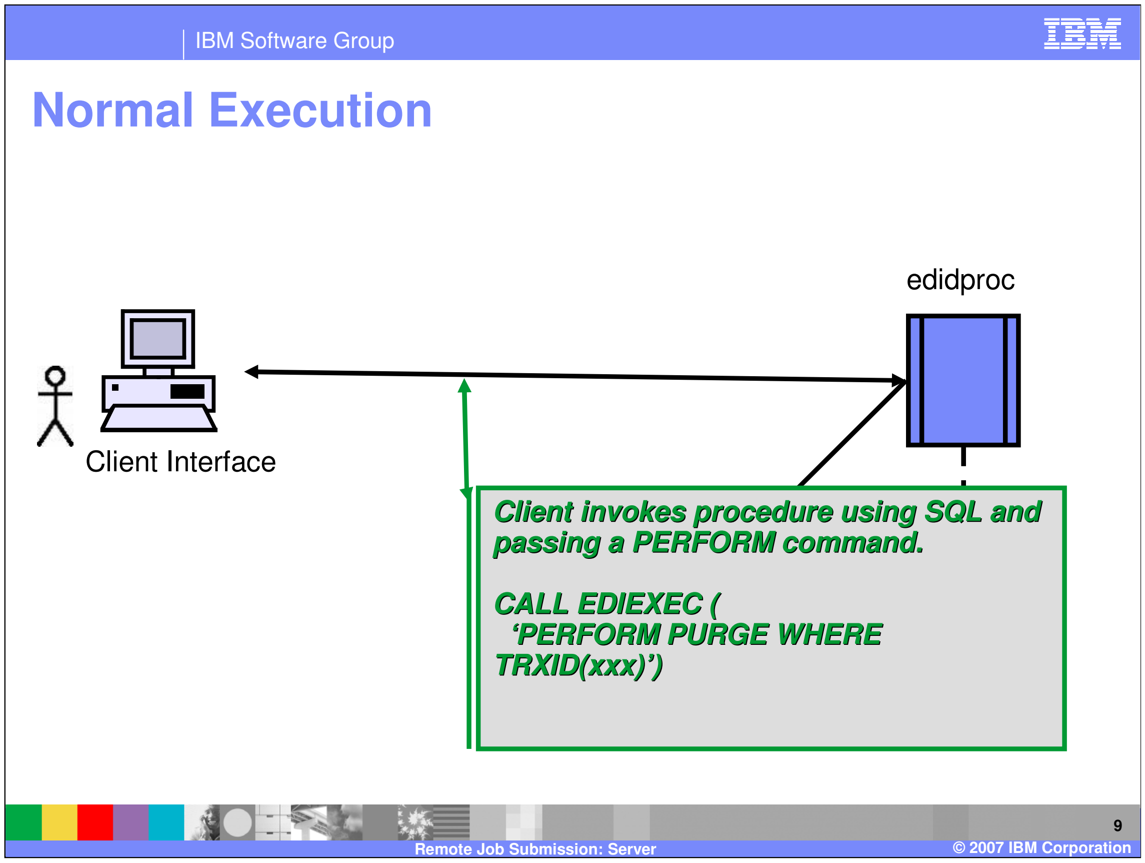
The other components represented here are the processes and the shared DB2 entities. These are discussed as part of the normal execution scenario. The red box symbolized the DB2 execution environment for stored procedures discussed in more detail in the “setup” section.



## Normal Execution

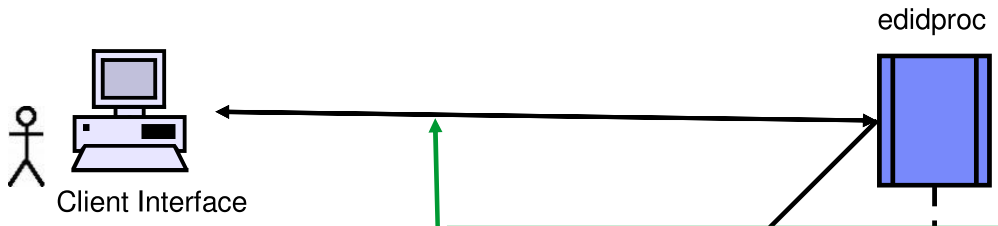


Normal execution is always initiated by an actor working on the WDI Client. The various areas within the Client including Document Store list windows, Service Profile edit windows and the Submissions functional area follow an identical process in executing remote functions.



IBM Software Group

## Normal Execution



***Client invokes procedure using SQL and passing a PERFORM command.***

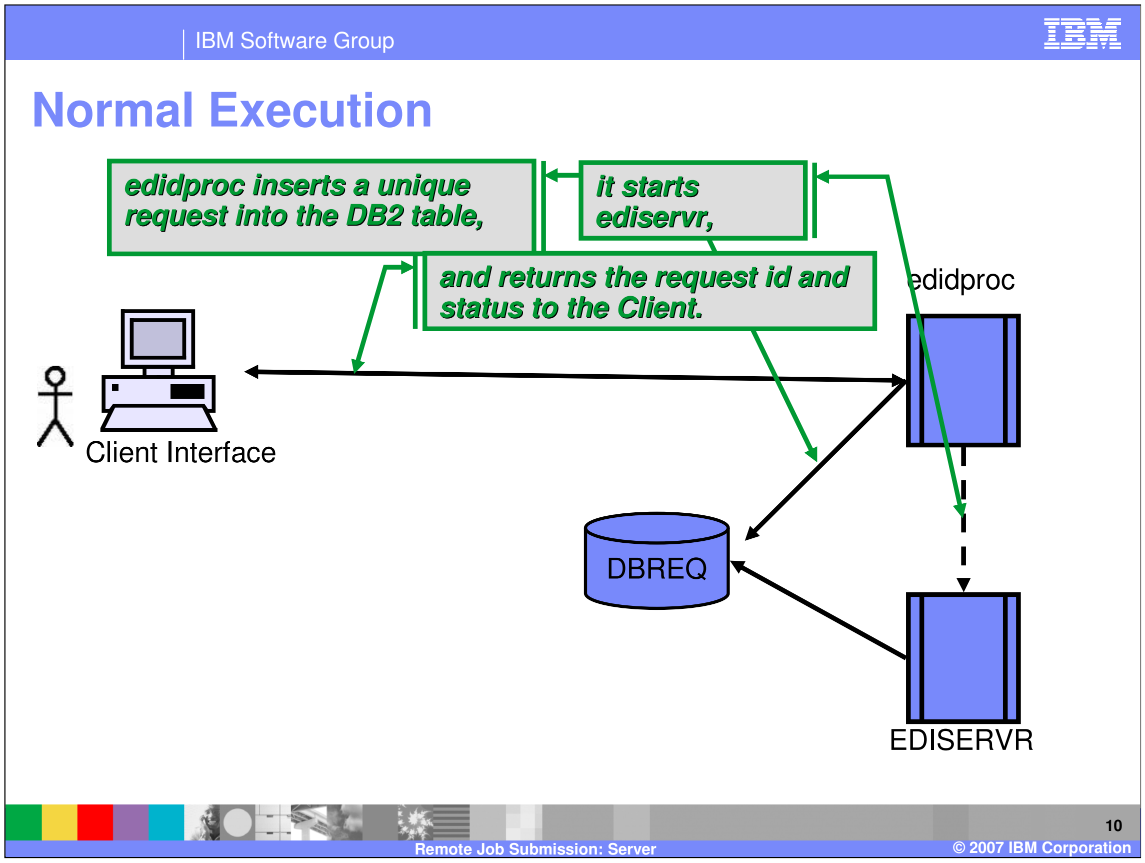
***CALL EDIEXEC ( 'PERFORM PURGE WHERE TRXID(xxx)')***

9

Remote Job Submission: Server

© 2007 IBM Corporation

The first step is to construct a complete and valid command script. The client sends this to remote stored procedure as a value in an SQL CALL statement. All calls are handled by the "edidproc" executable. This is distributed as a load module (LMOD) on z/OS, as a shared library on AIX and as a Dynamic Link Library (DLL) on Windows. In general discussion, "edidproc" is the stored procedure.



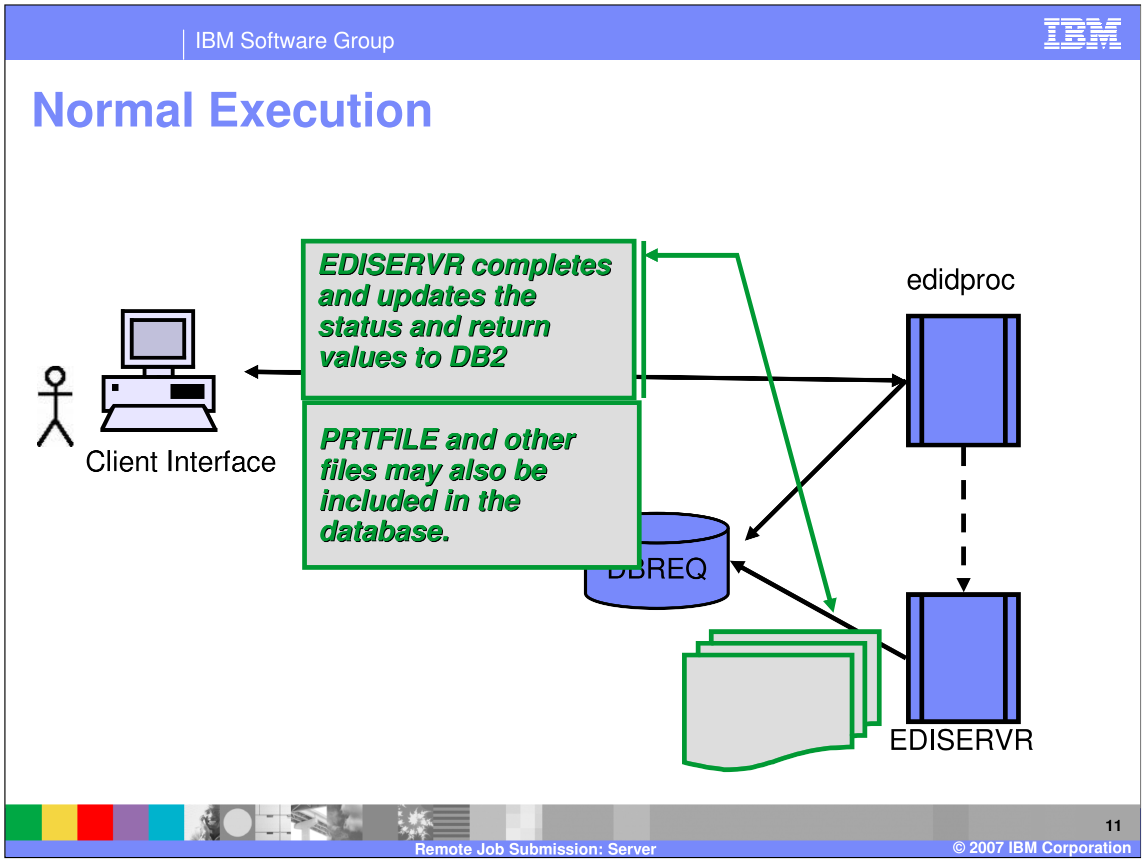
## Normal Execution

edidproc inserts a unique request into the DB2 table,
it starts ediservr,
and returns the request id and status to the Client.
edidproc
Client Interface
DBREQ
EDISERVR
10
Remote Job Submission: Server
© 2007 IBM Corporation

The second step is placing the request into a persistent DB2 entity. This assigns a unique ID number to the request for status and tracking. In some cases, “edidproc” may complete processing here and return control to the Client.

In the next step, “edidproc” starts an independently executing instance of the command interpreter, “ediservr,” passing it the completed command script. All scripts include syntax to tell “ediservr” its unique request ID. The completed script also directs the Audit Log’s PRTFILE DD to the database. For debugging, “edidproc” may wait for this task to complete, but generally it continues once the new process starts.

As it terminates, the stored procedure returns the new request ID and an overall status to the invoking Client. This frees the Client thread from the ODBC call and allows the terminal user to move on to other functions.

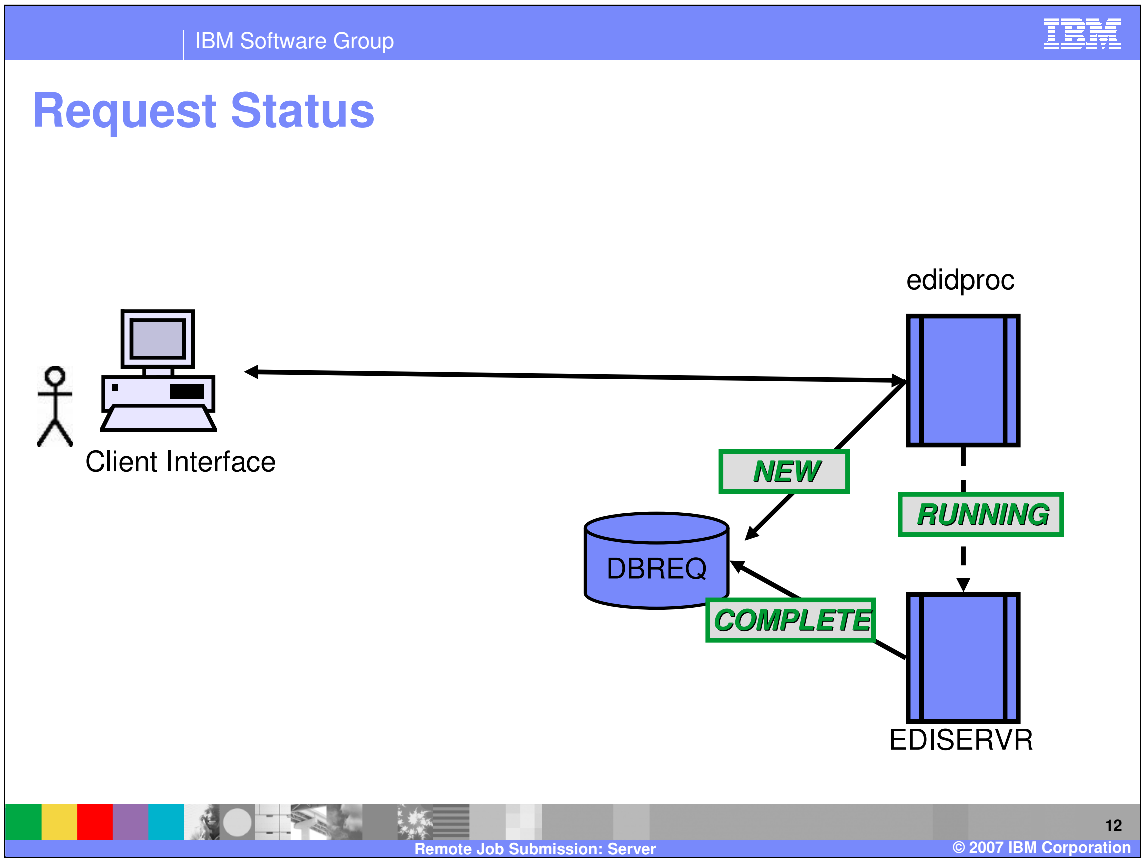


IBM Software Group

## Normal Execution

EDISERVR completes and updates the status and return values to DB2
PRTFILE and other files may also be included in the database.
edidproc
Client Interface
DBREQ
EDISERVR
11
Remote Job Submission: Server
© 2007 IBM Corporation

When it completes, “ediservr” updates the stored request with status information from the actual function performed. One or more outputs may have been directed to the data base during processing, and these are visible to the Client user. This output is limited to character based data such at the Audit Log. Non character data could easily be misinterpreted as it converts from the code page of the DB2 node to the Unicode database and finally to the code page used by the Windows Client.



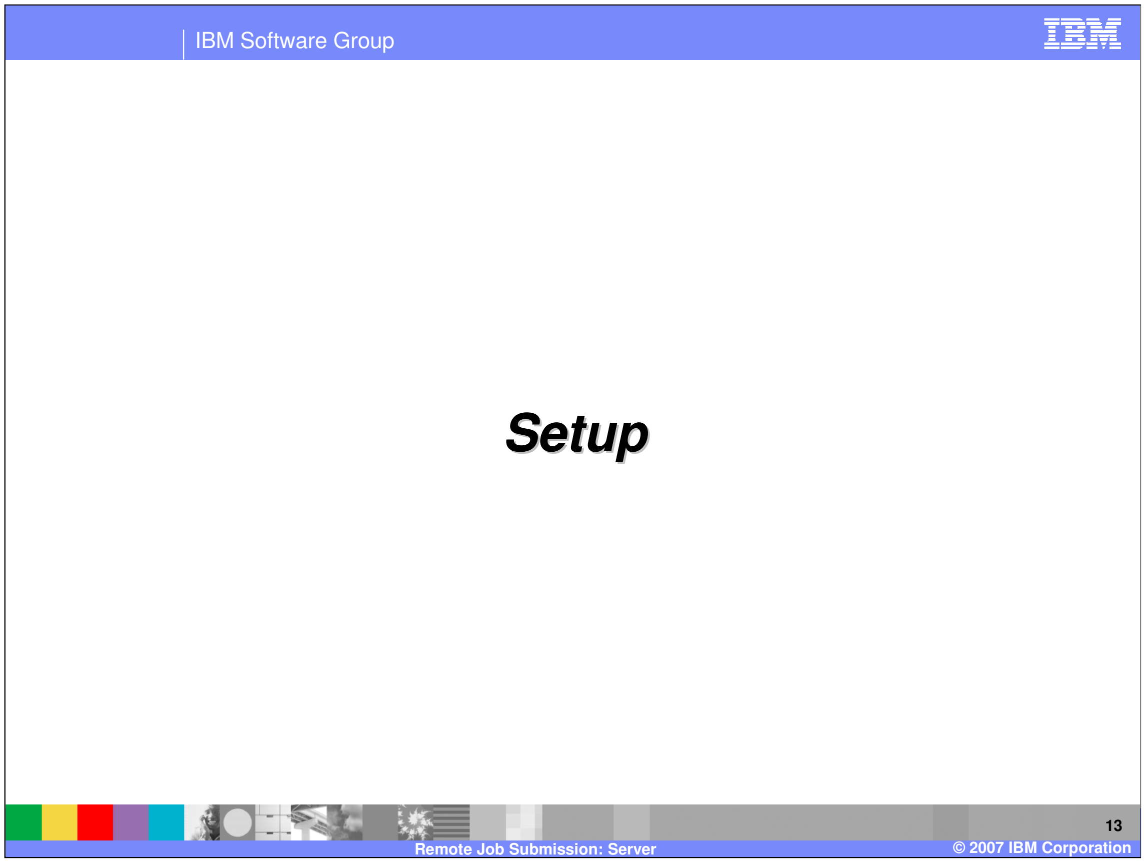
Ad the request flows through this process, its status changes within the persistent database row. The status is viewable through the Client's "submissions" functional area.

All requests start out on the database in New status.

When the request script is successfully passed to an instance of the server, it changes to Running.

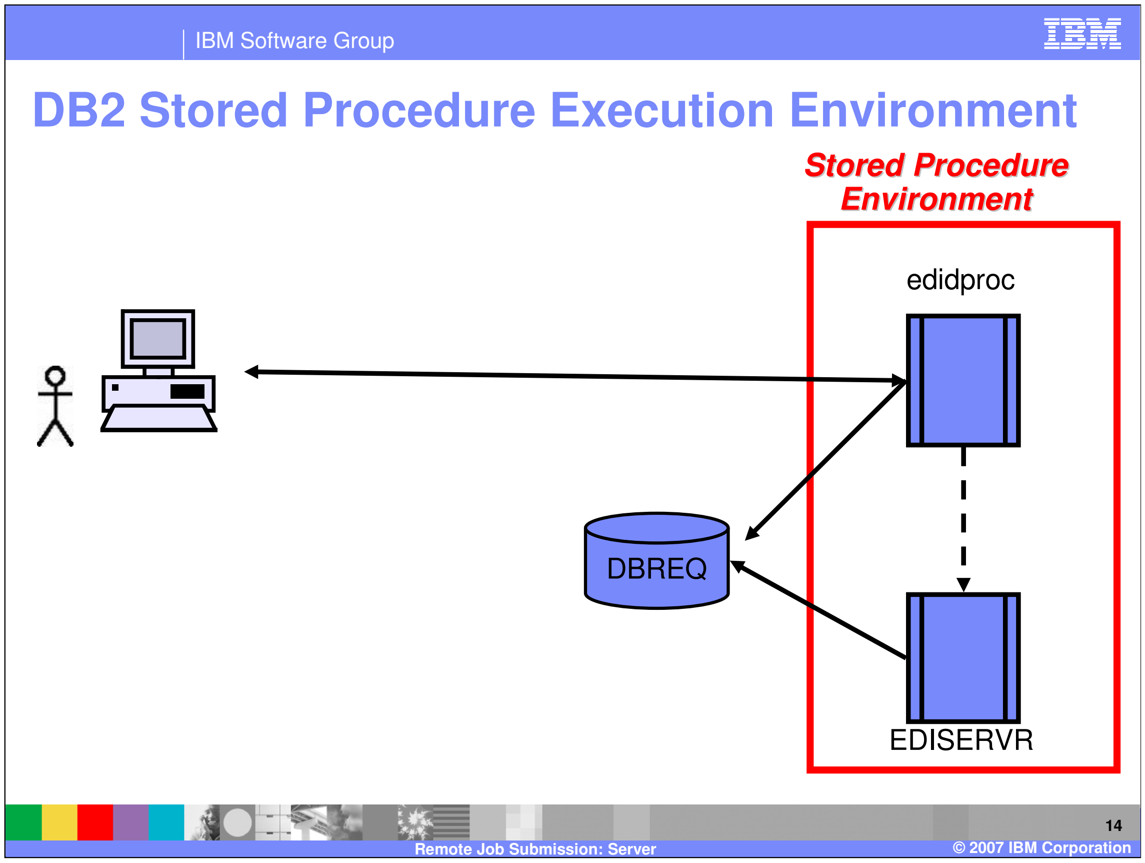
When the server is done, it changes the status to Complete.

Long running tasks may stay in Running state for some time, but a request that stays in this state may indicate some problem during processing.



## Setup

This section discusses install time issues specific to executing remote commands.


IBM Software Group

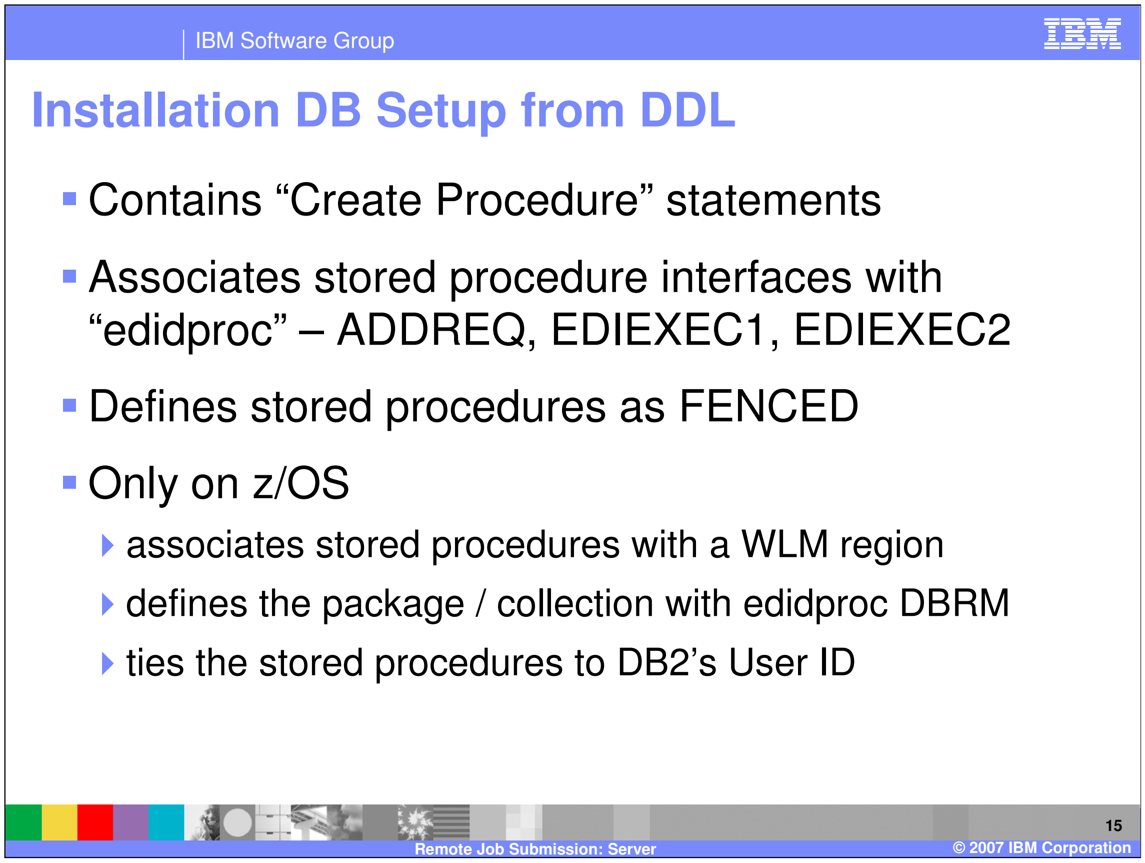
## DB2 Stored Procedure Execution Environment

14

Remote Job Submission: Server

© 2007 IBM Corporation

Remote commands execute as a DB2 stored procedure.  Since this is new for WDI version 3.3, there is some minor configuration necessary to allow DB2 access to the load modules involved.



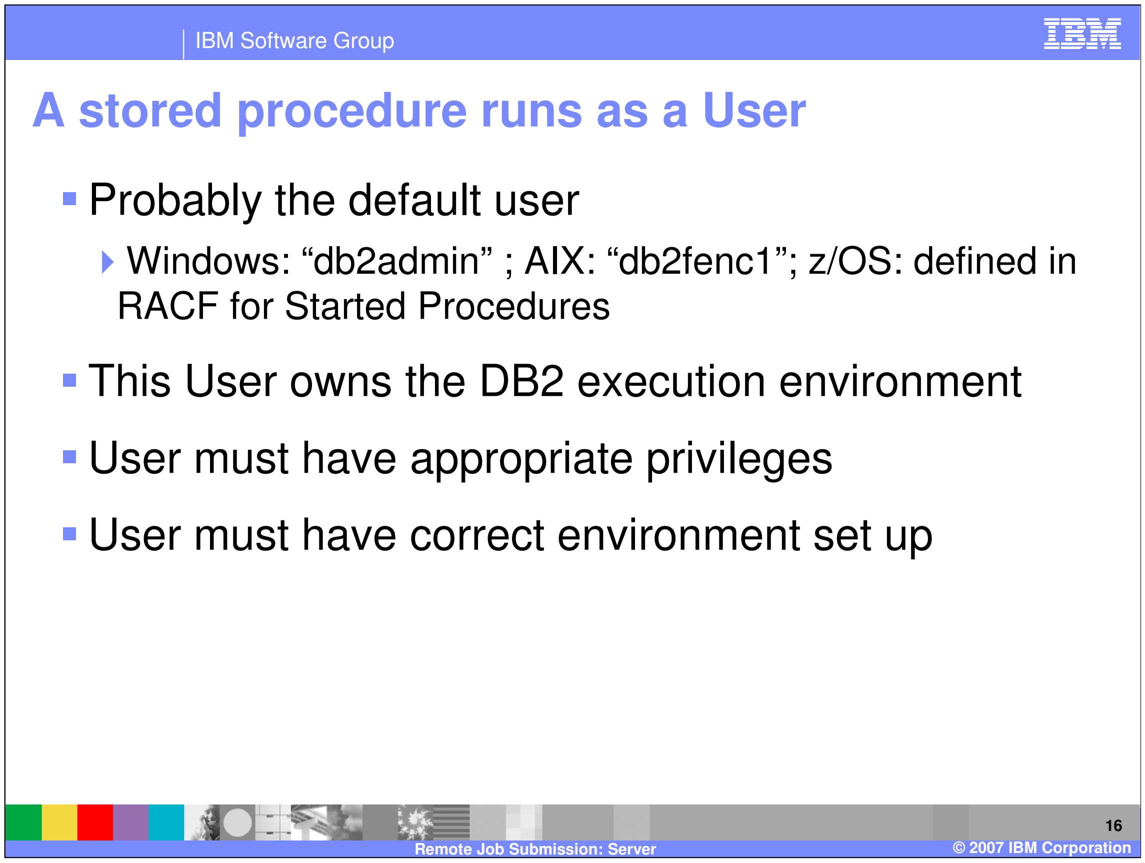
## Installation DB Setup from DDL

- Contains "Create Procedure" statements
- Associates stored procedure interfaces with "edidproc" – ADDREQ, EDIEXEC1, EDIEXEC2
- Defines stored procedures as FENCED
- Only on z/OS
  - associates stored procedures with a WLM region
  - defines the package / collection with edidproc DBRM
  - ties the stored procedures to DB2's User ID

The DDL shipped with the product contains commands that define DB2 stored procedures. There are three definitions, though each of them is implemented using the single "edidproc" load module as noted earlier. The ADDREQ definition allows the Client to create a New request. EDIEXEC1 executes an already stored request while EDIEXEC2 performs adding and executing in a single step.

For Open platforms, these definitions are FENCED telling DB2 that they should be run in an isolated process environment with no ability to damage DB2's internal storage. This is a slower but safer way to execute.

Definitions on z/OS contain the name of the Workload Manager administered address space used to execute WDI. WDI can not execute in a DB2 managed stored procedure address space. The definition also contains the name of the DB2 package that contains the SQL used by "edidproc." This package collection does not need to contain all of the access information stored in the "plan" used to execute WDI.

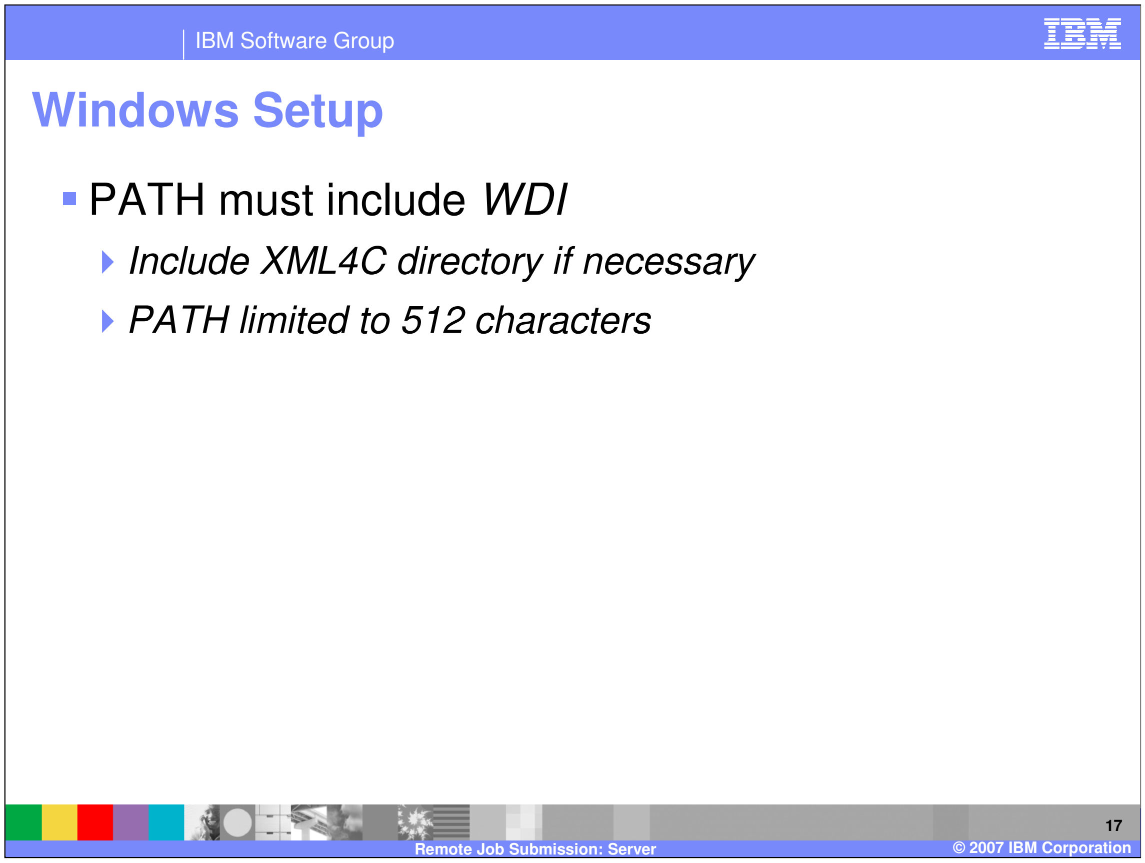


## A stored procedure runs as a User

- Probably the default user
  - Windows: “db2admin” ; AIX: “db2fenc1”; z/OS: defined in RACF for Started Procedures
- This User owns the DB2 execution environment
- User must have appropriate privileges
- User must have correct environment set up

It is important to remember that the stored procedure runs using the user credentials inherited from the DB2 execution and not with the credentials used by the user connecting to the DB2 node.  On Windows, the user is determined at installation time and may change if the node is reconfigured.  AIX always uses the pseudo user “db2fenc1.”  On z/OS, the user is configured using security tools like RACF.

The user executing the stored procedure must have access to any files or other controlled resources.

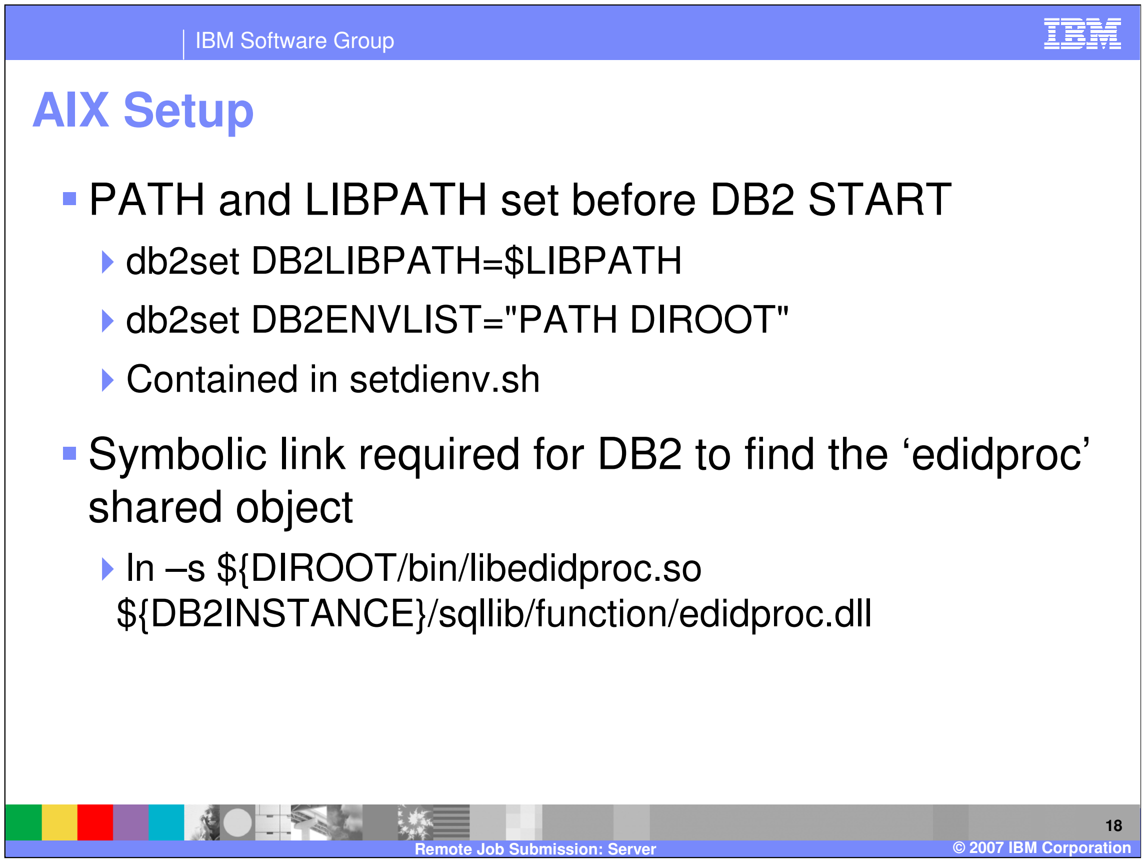


## Windows Setup

- PATH must include *WDI*
  - *Include XML4C directory if necessary*
  - *PATH limited to 512 characters*

The Windows stored procedure user must have the path for WDI and XML4C libraries within the first 512 characters of the path name. Installation may append these directories to the PATH, so a new installation may have to manually move this directory up the PATH.

Note well that any changes made to the environment take effect only after the database restarts.


IBM Software Group

## AIX Setup

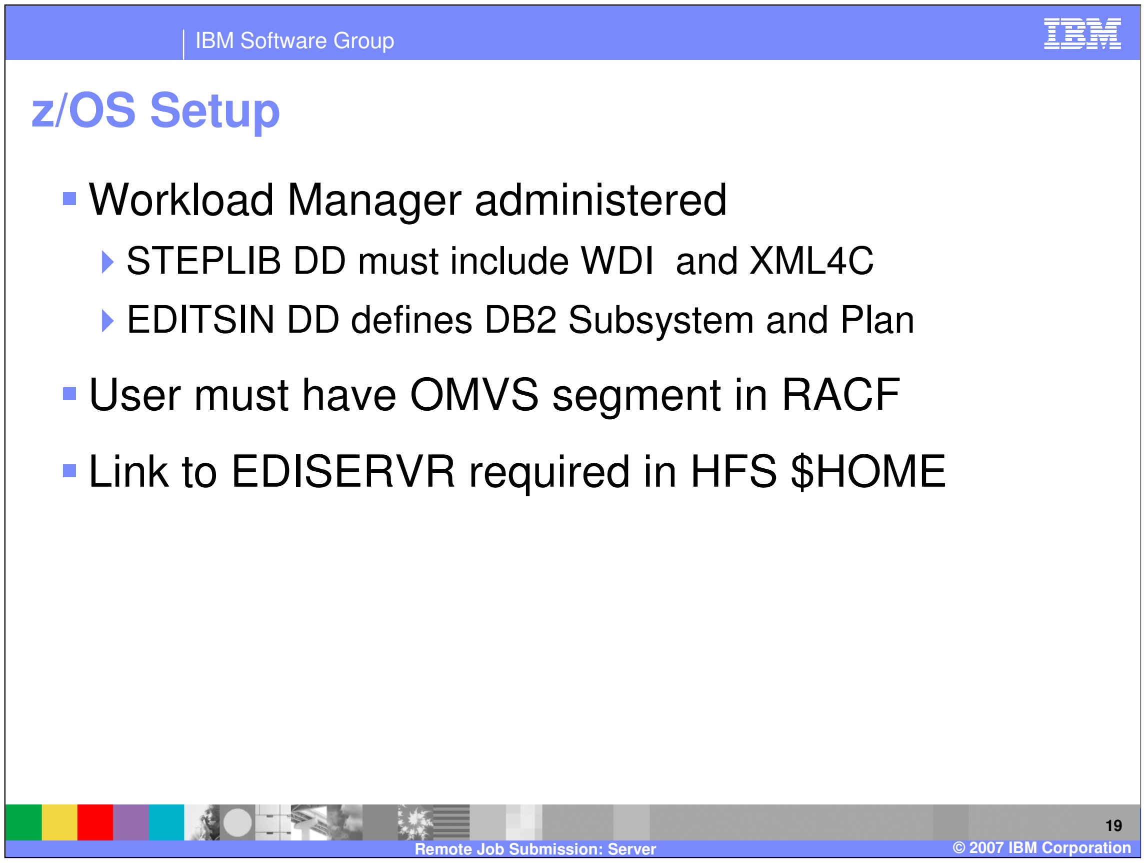
- PATH and LIBPATH set before DB2 START
  - db2set DB2LIBPATH=$LIBPATH
  - db2set DB2ENVLIST="PATH DIROOT"
  - Contained in setdienv.sh
- Symbolic link required for DB2 to find the 'edidproc' shared object
  - ln –s ${DIROOT/bin/libedidproc.so ${DB2INSTANCE}/sqllib/function/edidproc.dll

18

Remote Job Submission: Server

© 2007 IBM Corporation

AIX requires that both the PATH and LIBPATH environment variable contain a reference to the installation path for WDI. These must be exported to DB2's variable list using the 'db2set' command as noted and contained in setdienv.sh shell script distributed with the product. Additionally, there must be a symbolic link in the ../function subdirectory under the installation home for the DB2 instance. This is the only directory searched for fenced stored procedures. This link also maps the name of the shared object library, libedidproc.so, to the name 'edidproc.dll' contained in the DDL.



## z/OS Setup

- Workload Manager administered
  - STEPLIB DD must include WDI  and XML4C
  - EDITSIN DD defines DB2 Subsystem and Plan
- User must have OMVS segment in RACF
- Link to EDISERVR required in HFS $HOME

As with any DB2 change on z/OS, it is important to involve a knowledgeable DBA in the installation process.  This description does not cover all of the SAF and other changes required to configure a host DB2 node.  Rather, it focuses on the changes specific to WDI.

The z/OS Workload Manager administers the address space used to execute the stored procedure.  This address space is started using a JCL PROC with the same name as the WLM argument specified in the DDL.  This PROC must allocate the STEPLIB to include both the WDI load library and the load library for XML4C.  Additionally, it must allocate an EDITSIN DD that is used by the product to attach back to DB2 from the WLM address space.  This connection must be made using a valid plan and the RRSAF attachment facility.  The contents of an EDITSIN parm might look like this.

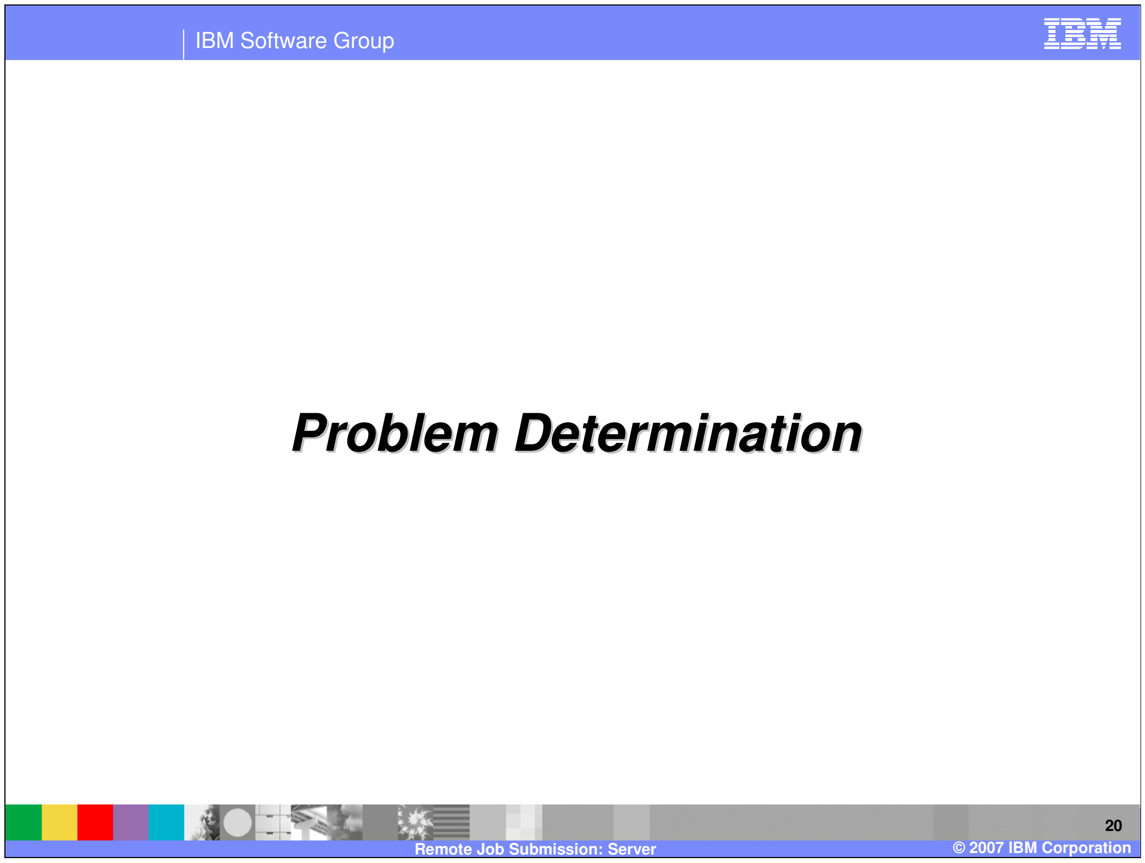
```
PLAN(EDI52ACC) CAF(8) SYSTEM(DB2X)
```

This DD is documented in the WebSphere® Data Interchange Programmer's Reference, and it must specify "CAF(8)" to invoke RRSAF.

The ID associated with the WLM address space must have an OMVS segment defining a home directory, and the must be an external link in this HFS directory for EDISERVR.  The syntax for this might look like this.

```
ln -e EDISERVR /home/user1/EDISERVR
```

The name must be upper case.


IBM Software Group

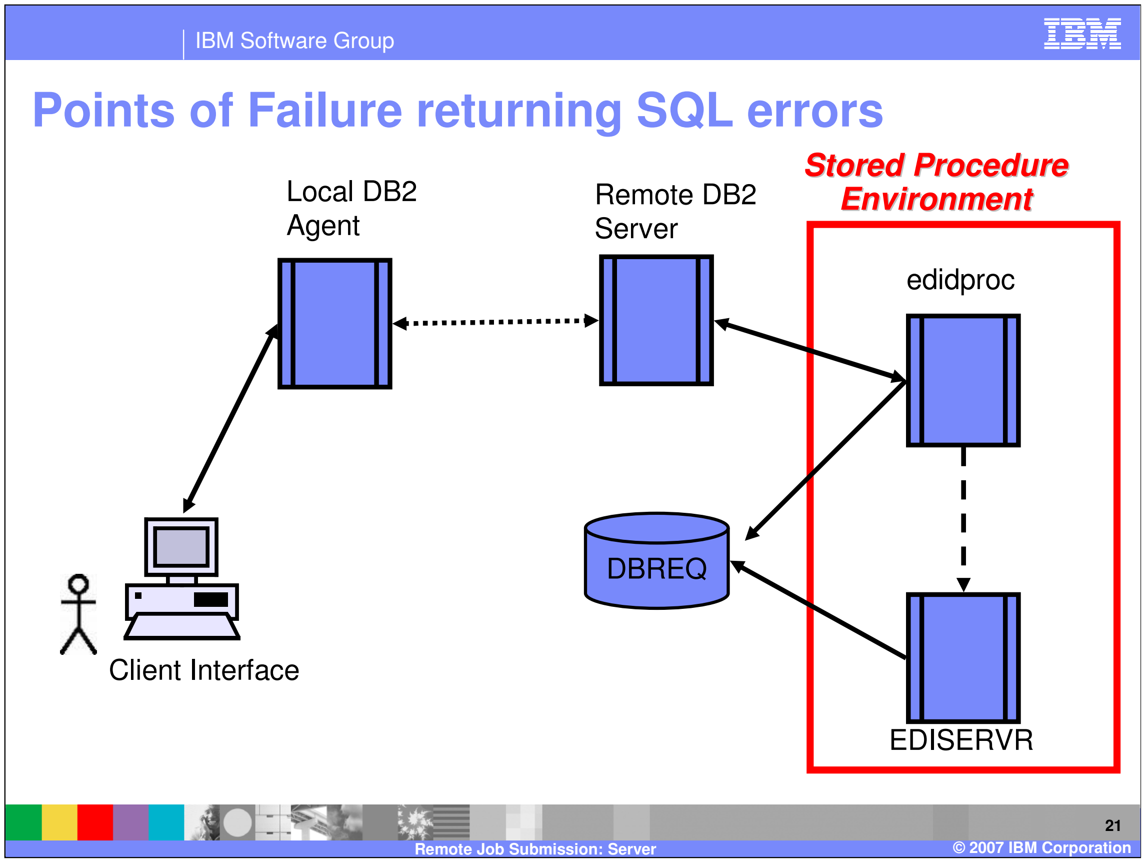
## Problem Determination

20

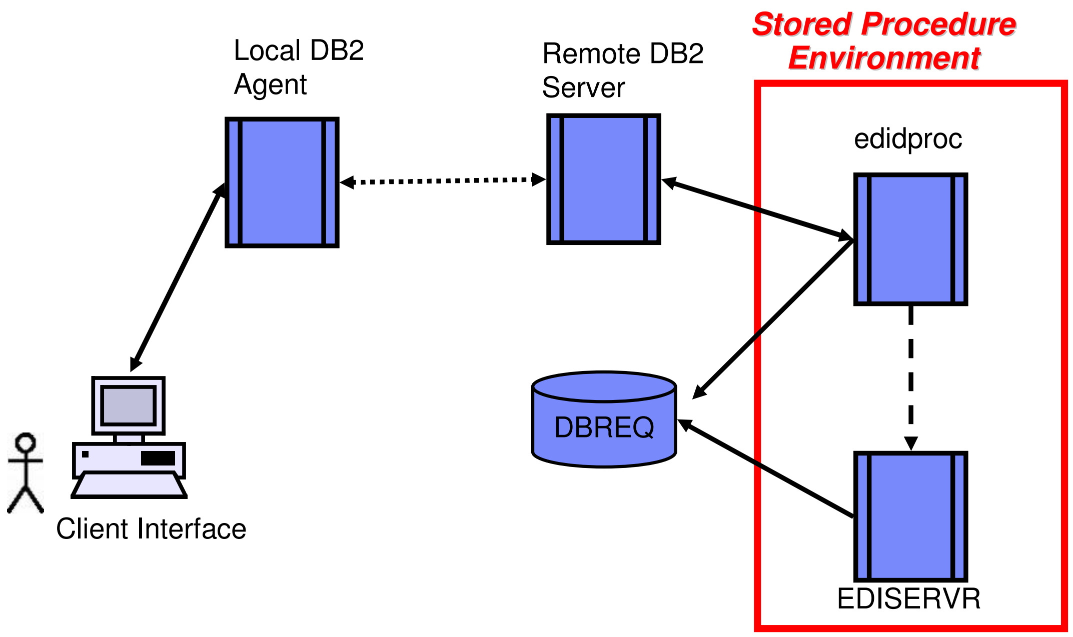
Remote Job Submission: Server

© 2007 IBM Corporation

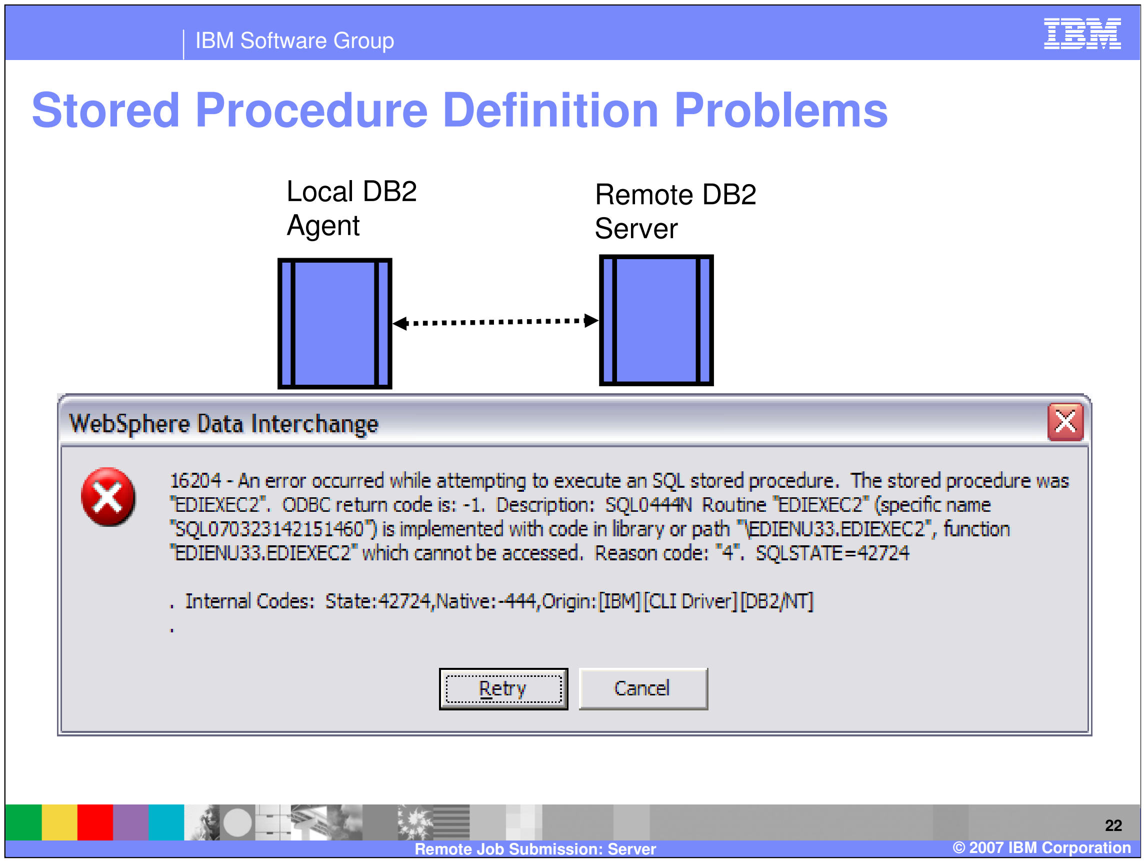
This section discusses ways of finding out what went wrong in the event of an error.



## Points of Failure returning SQL errors



Excluding code issues, the Remote Command process could fail at any point where processes interact.



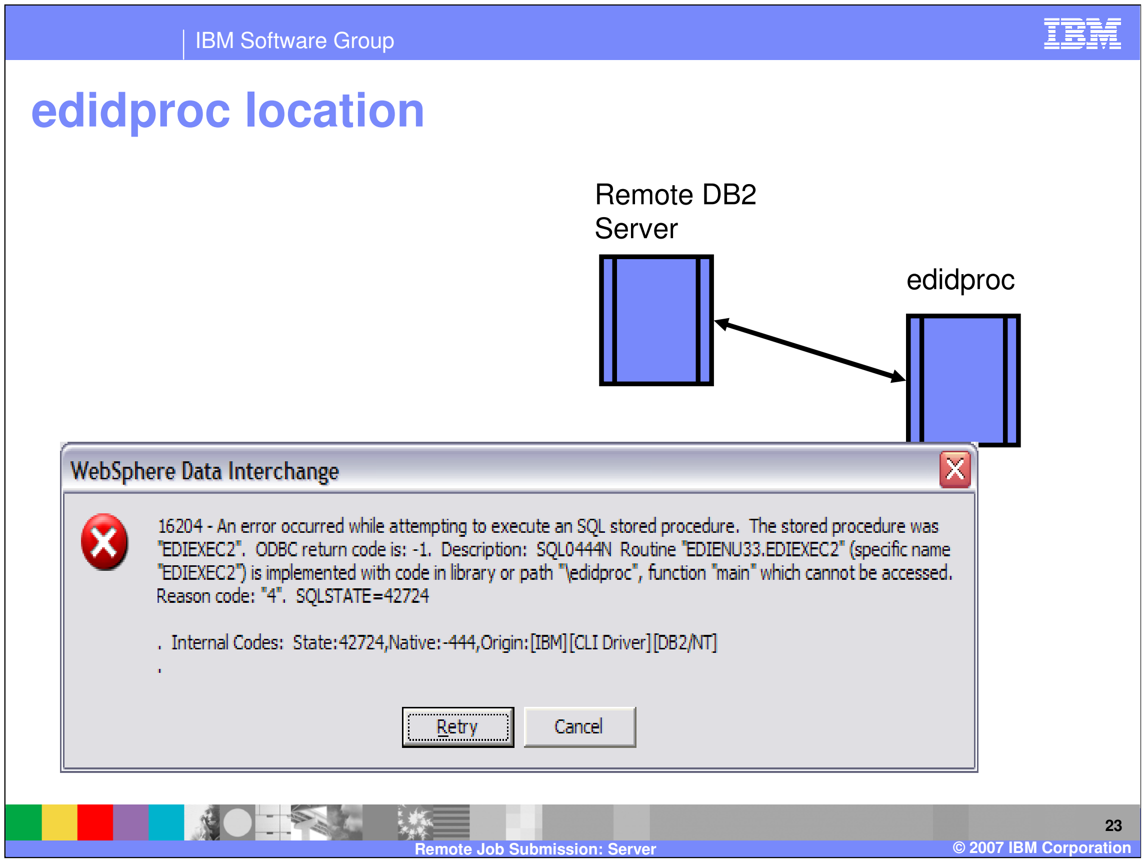
## Stored Procedure Definition Problems

16204 - An error occurred while attempting to execute an SQL stored procedure. The stored procedure was "EDIEXEC2". ODBC return code is: -1. Description: SQL0444N Routine "EDIEXEC2" (specific name "SQL070323142151460") is implemented with code in library or path "\EDIENU33.EDIEXEC2", function "EDIENU33.EDIEXEC2" which cannot be accessed. Reason code: "4". SQLSTATE=42724

. Internal Codes: State:42724,Native:-444,Origin:[IBM][CLI Driver][DB2/NT]
.

During installation and setup, it is not unusual to see a message box on the Client lake that displayed here. The specific failure code is SQL 444 that indicates some failure invoking a stored procedure. This may indicate that the stored procedure definition itself is not accessible to the current user or that it is not defined correctly.

The message below actually indicates a midding Stored Prodecure definition. Contrast the function name here with that on the next screen.



IBM Software Group

## edidproc location

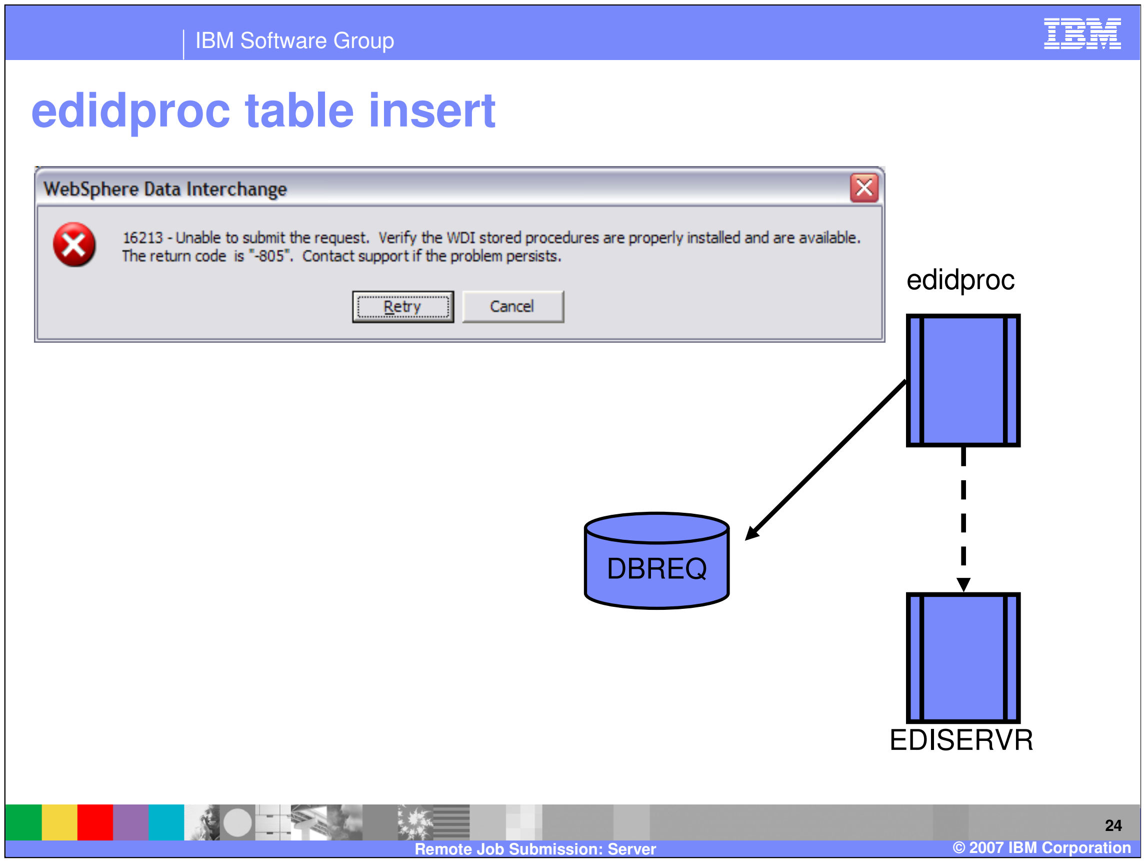
```
16204 - An error occurred while attempting to execute an SQL stored procedure.  The stored procedure was
"EDIEXEC2".  ODBC return code is: -1.  Description:  SQL0444N  Routine "EDIENU33.EDIEXEC2" (specific name
"EDIEXEC2") is implemented with code in library or path "\edidproc", function "main" which cannot be accessed.
Reason code: "4".  SQLSTATE=42724

. Internal Codes:  State:42724,Native:-444,Origin:[IBM][CLI Driver][DB2/NT]
.
```

23

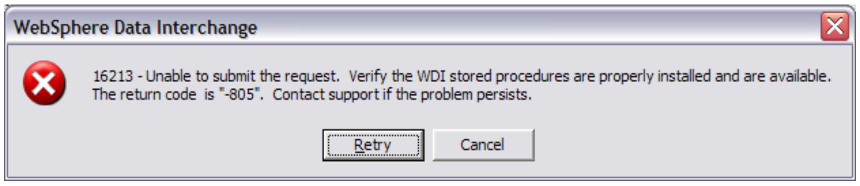
Remote Job Submission: Server

© 2007 IBM Corporation

A similar error could be produced in the DB2 instance is not able to load the "edidproc.dll" load module.  This may be the result of a missing or incorrect symbolic link, an incorrect PATH or permissions to execute the package..

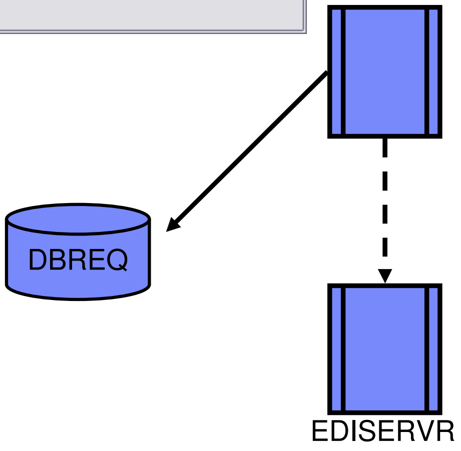


## edidproc table insert



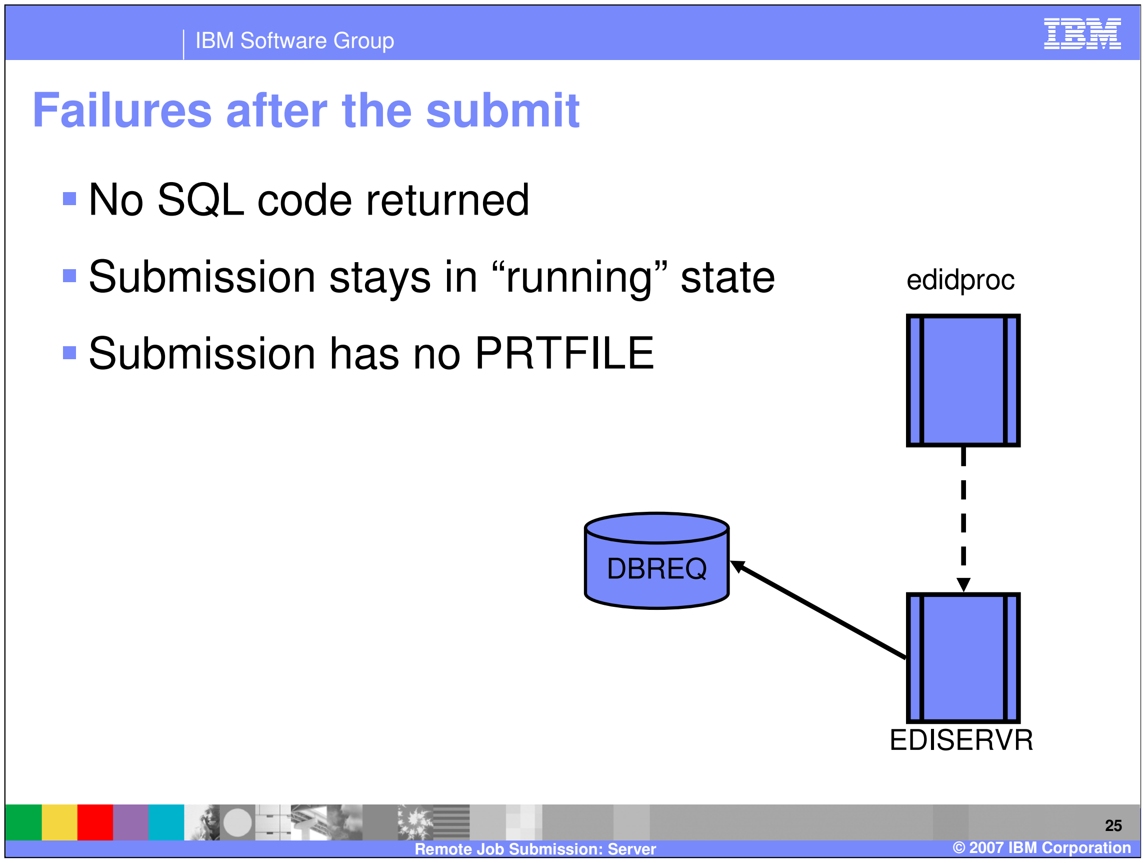
WebSphere Data Interchange

16213 - Unable to submit the request. Verify the WDI stored procedures are properly installed and are available. The return code is "-805". Contact support if the problem persists.



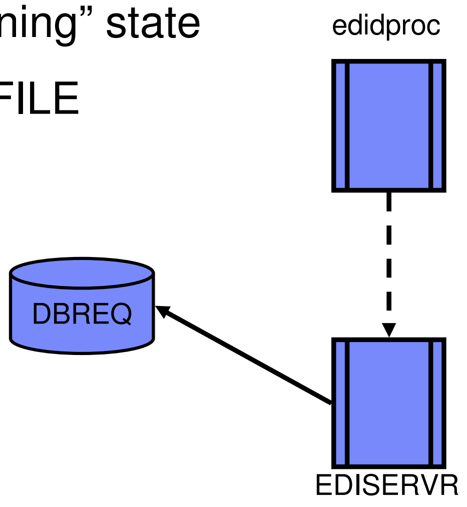
As the stored procedure executes its own SQL, any errors returned by DB2 are transferred to the Client. Here, a -805 because of a package mismatch is displayed on the Client using message 16213. This is the last point to return status to the Client.

This same message may return non-DB2 error code if there is a problem starting EDISERVR. Should this happen, the error code returned by the native operating system will appear in the message. This could indicate a failure to find the load module, the symbolic link or one of the supporting libraries. It could also indicate a permissions problem.



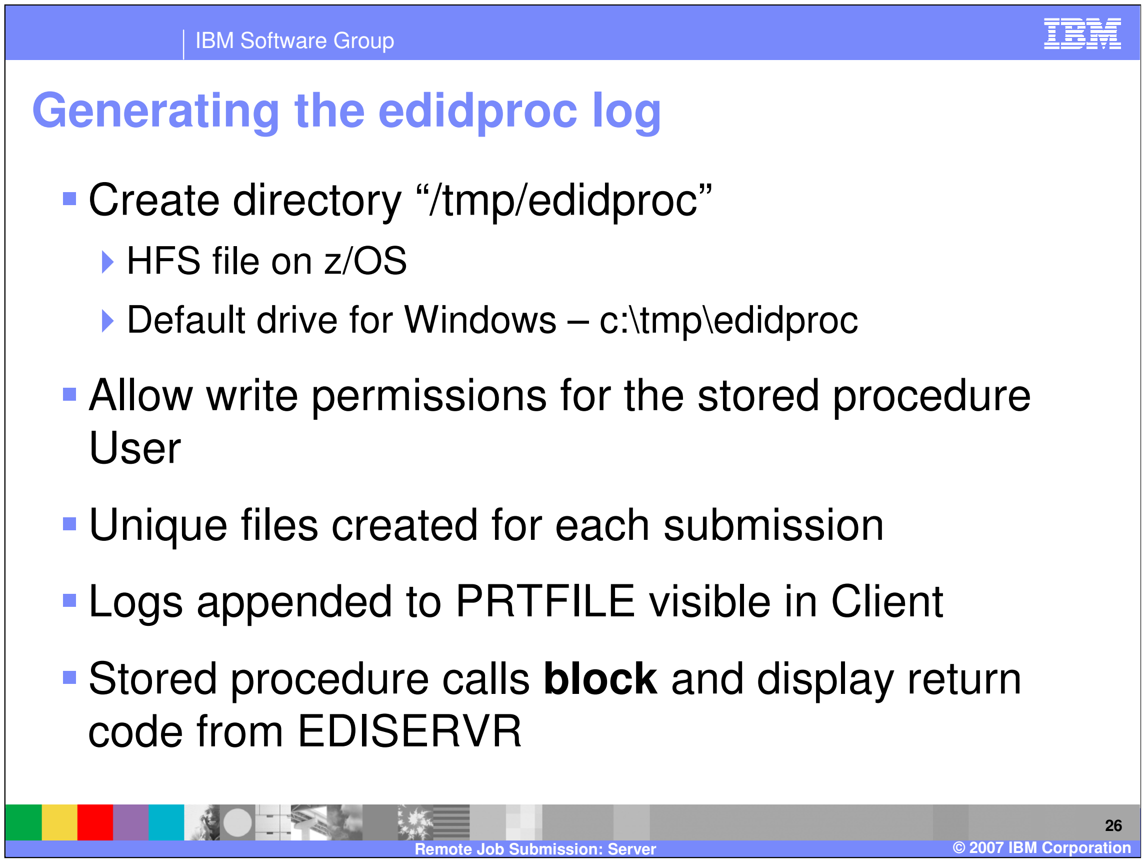
## Failures after the submit

- No SQL code returned
- Submission stays in “running” state
- Submission has no PRTFILE



After this point in the processing, no errors are returned to the Client. The remote request should now appear in the “submissions” area of the client with a New or Running status. If the request remains in this state for an extended period, it may indicate that there is a problem.

No Audit Log appears in the Client for many of these type of failures. You may be able to find additional information by using the Clients “Data References” to direct the PRTFILE to a physical file rather than defaulting it to the database.

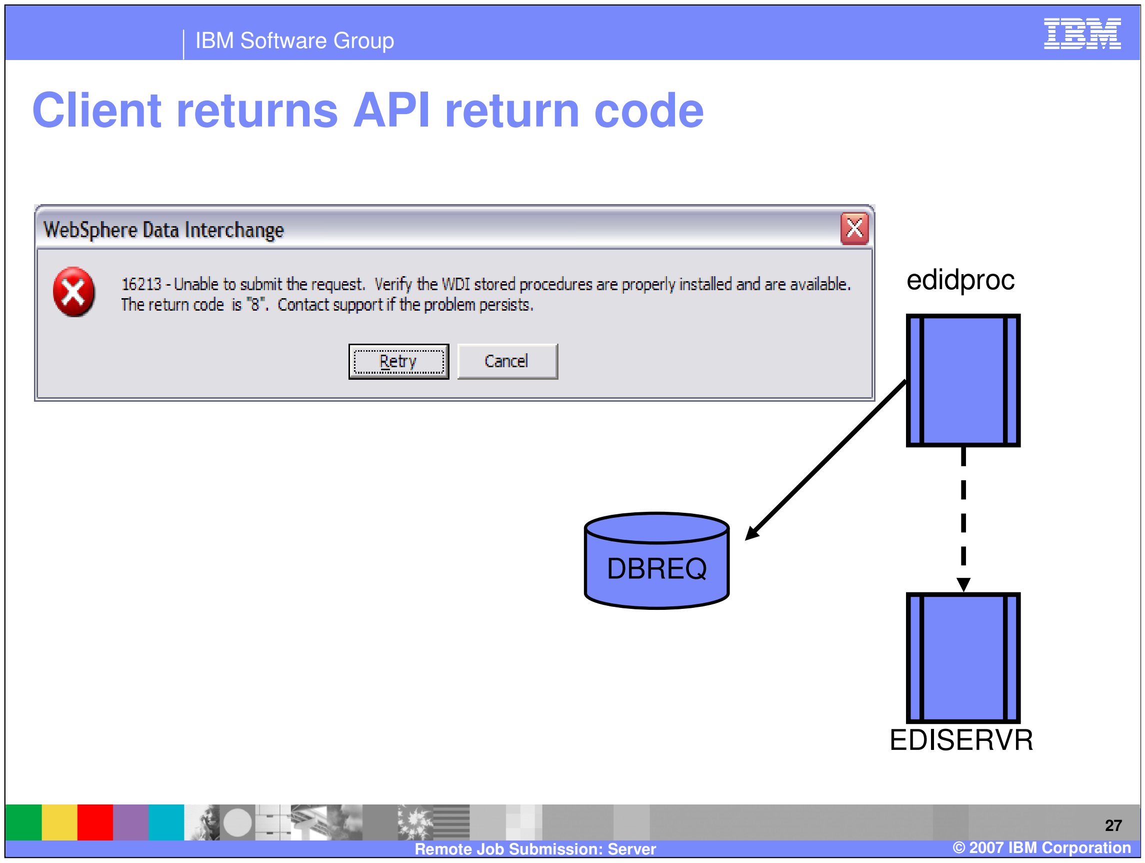


## Generating the edidproc log

- Create directory “/tmp/edidproc”
  - HFS file on z/OS
  - Default drive for Windows – c:\tmp\edidproc
- Allow write permissions for the stored procedure User
- Unique files created for each submission
- Logs appended to PRTFILE visible in Client
- Stored procedure calls **block** and display return code from EDISERVR

To get a better idea of what is happening in the stored procedure, create a logging directory called “edidproc” in the “/tmp” file system.  Subsequent submissions will generate messages in a unique file within this directory as “edidproc” executes.  These logs can be viewed at the remote node, or they may be viewed as part of the PRTFILE within the “submissions” provided that processing completes.

While in this debugging mode, the stored procedure will always wait for  WDI to complete processing the request before returning control to the Client.  This long wait could cause Client screens to erase and could tie up a Client session indefinitely.  This allows the stored procedure to capture return codes from “ediservr.”  On z/OS, this will also capture any messages written to the JES log during execution.



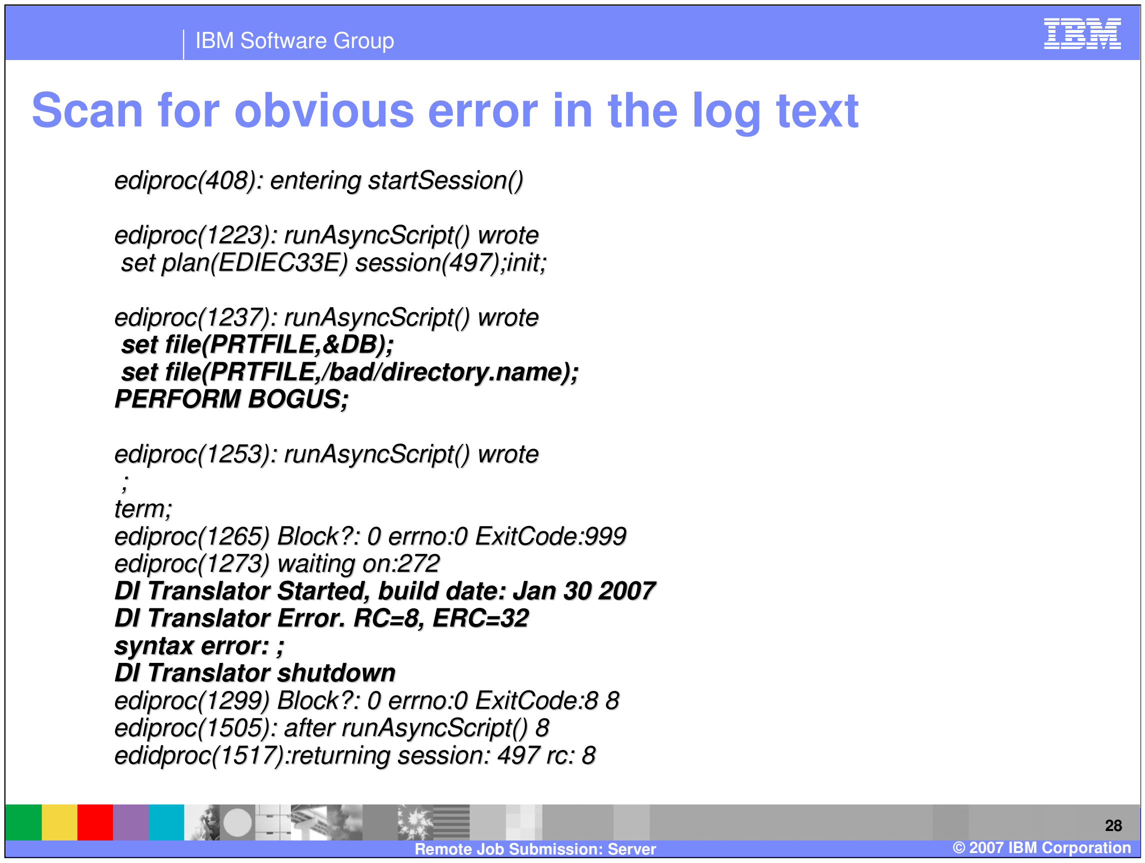
## Client returns API return code

WebSphere Data Interchange

16213 - Unable to submit the request. Verify the WDI stored procedures are properly installed and are available. The return code is "8". Contact support if the problem persists.

Retry
Cancel
edidproc
DBREQ
EDISERVR

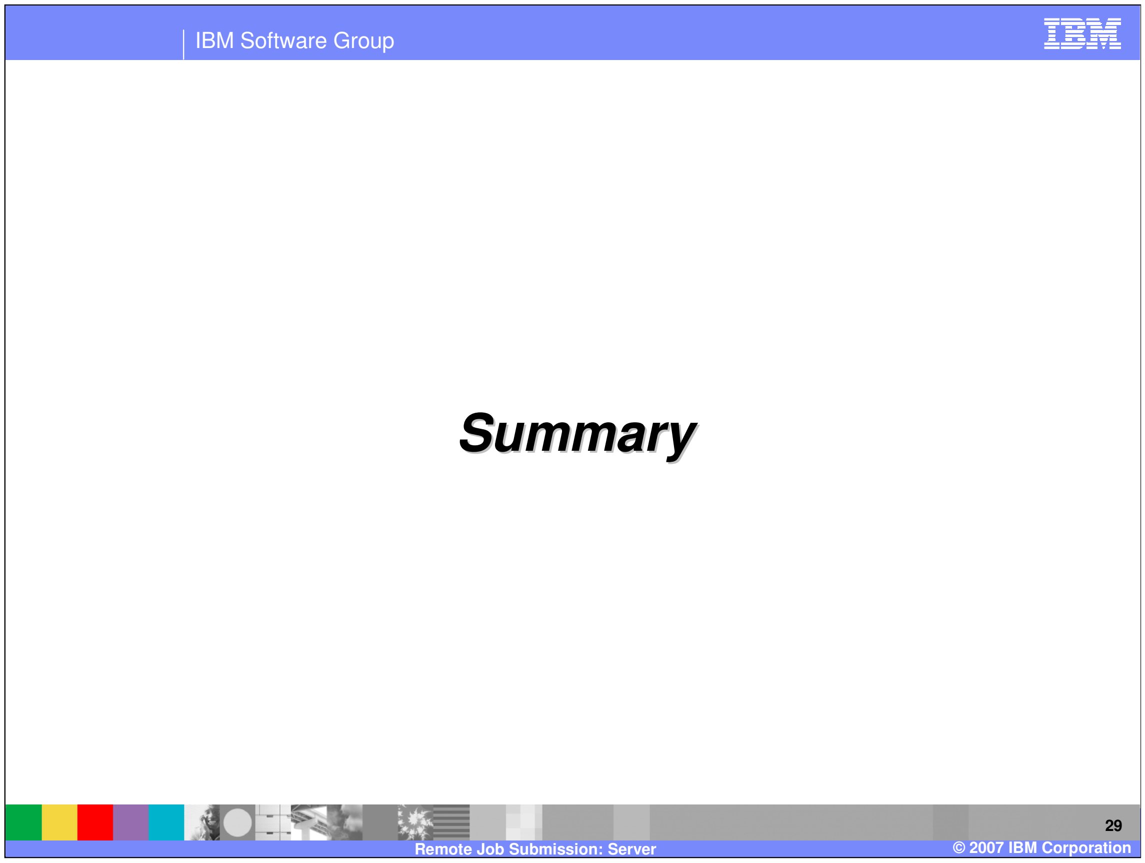
The Application Programming Interface (API) return code from the WDI server travels back to the Client when debugging.



## Scan for obvious error in the log text

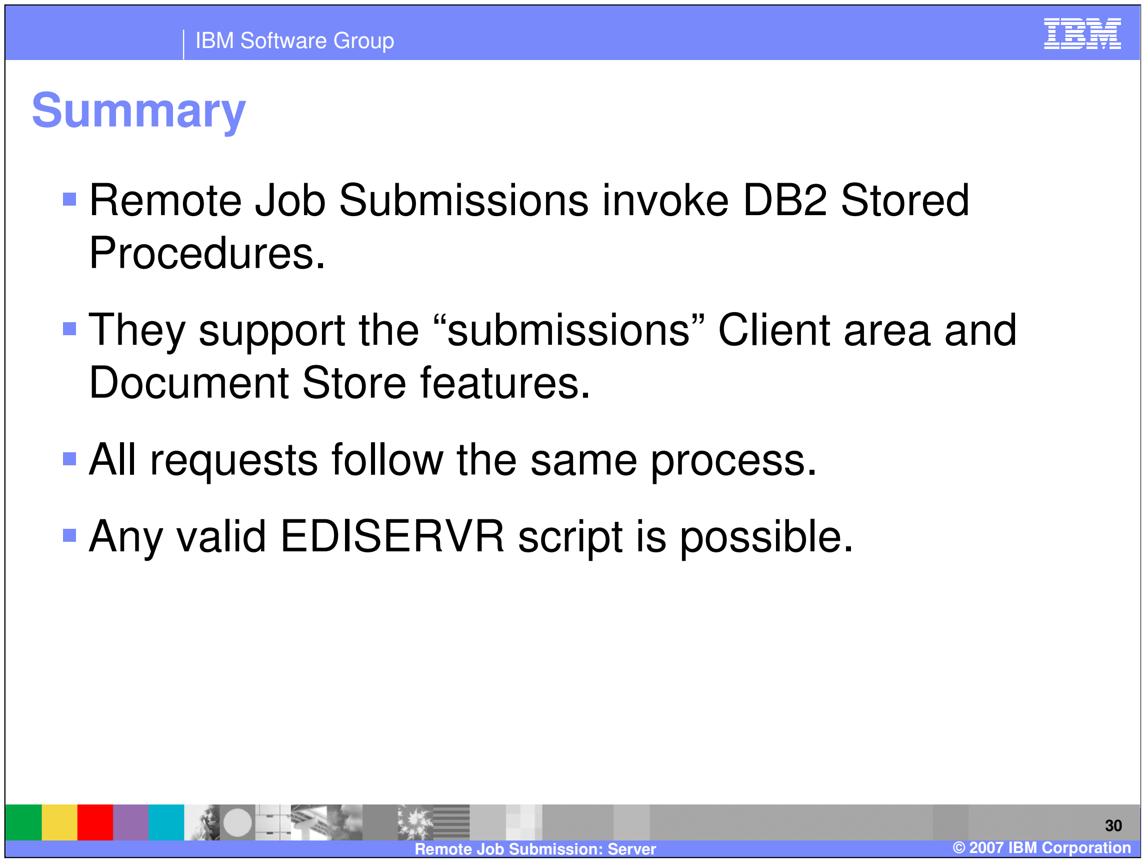
```
ediproc(408): entering startSession()

ediproc(1223): runAsyncScript() wrote
 set plan(EDIEC33E) session(497);init;

ediproc(1237): runAsyncScript() wrote
 set file(PRTFILE,&DB);
 set file(PRTFILE,/bad/directory.name);
PERFORM BOGUS;

ediproc(1253): runAsyncScript() wrote
 ;
term;
ediproc(1265) Block?: 0 errno:0 ExitCode:999
ediproc(1273) waiting on:272
DI Translator Started, build date: Jan 30 2007
DI Translator Error. RC=8, ERC=32
syntax error: ;
DI Translator shutdown
ediproc(1299) Block?: 0 errno:0 ExitCode:8 8
ediproc(1505): after runAsyncScript() 8
edidproc(1517):returning session: 497 rc: 8
```

The exact contents of the debug log are not documented here. They provide detailed information that may be of use to IBM support in performing problem determination. However, the general flow of the stored procedure should be echoed in this log, and you may be able to identify logical errors. In this case, there is a “set file” that directs the PRTFILE to an invalid directory. The log also shows the server starting and returning an extended return code 32 that indicates a fundamental initialization error. This is usually a failure to connect to the database manager or, as in this case, an inability to open the Audit Log.



## Summary

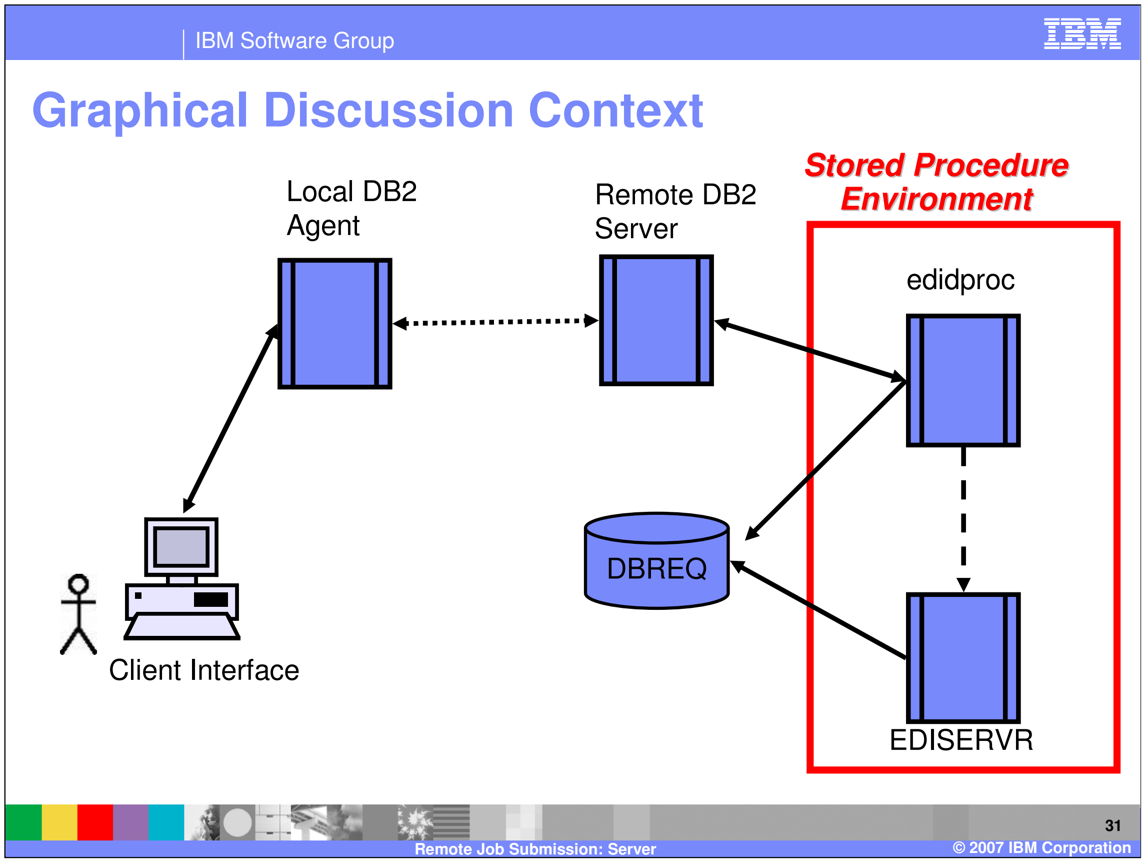
This reiterates the content of the prior presentation.



## Summary

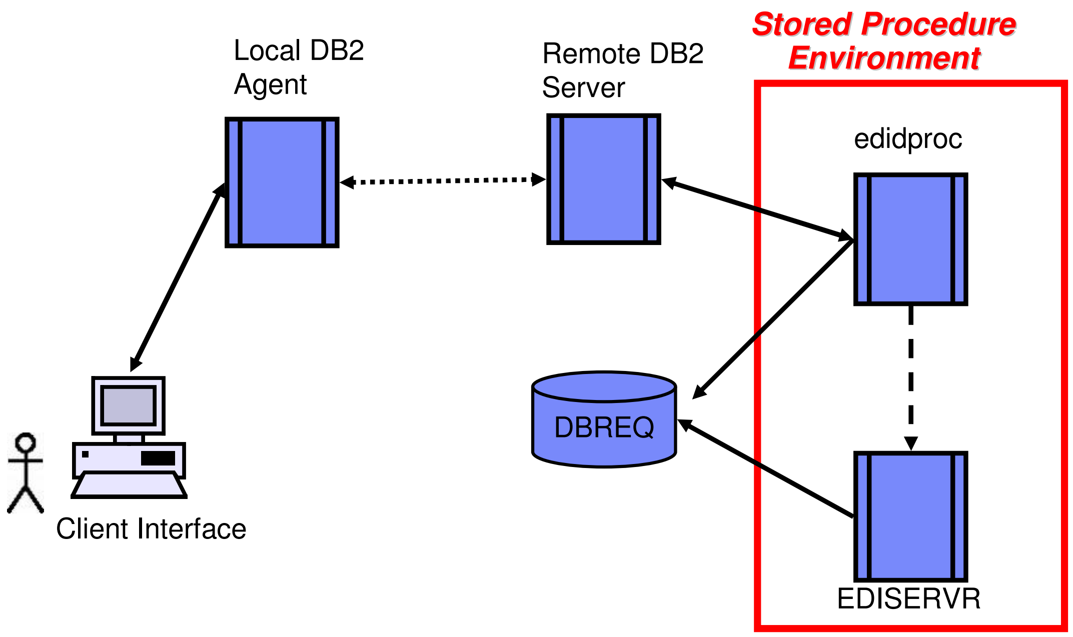
- Remote Job Submissions invoke DB2 Stored Procedures.
- They support the "submissions" Client area and Document Store features.
- All requests follow the same process.
- Any valid EDISERVR script is possible.

Client "submissions" are requests executed by a DB2 stored procedure. Though any PERFORM command may be run directly in the "submissions" functional area, the Document Store features construct and execute standard requests.



IBM Software Group

## Graphical Discussion Context

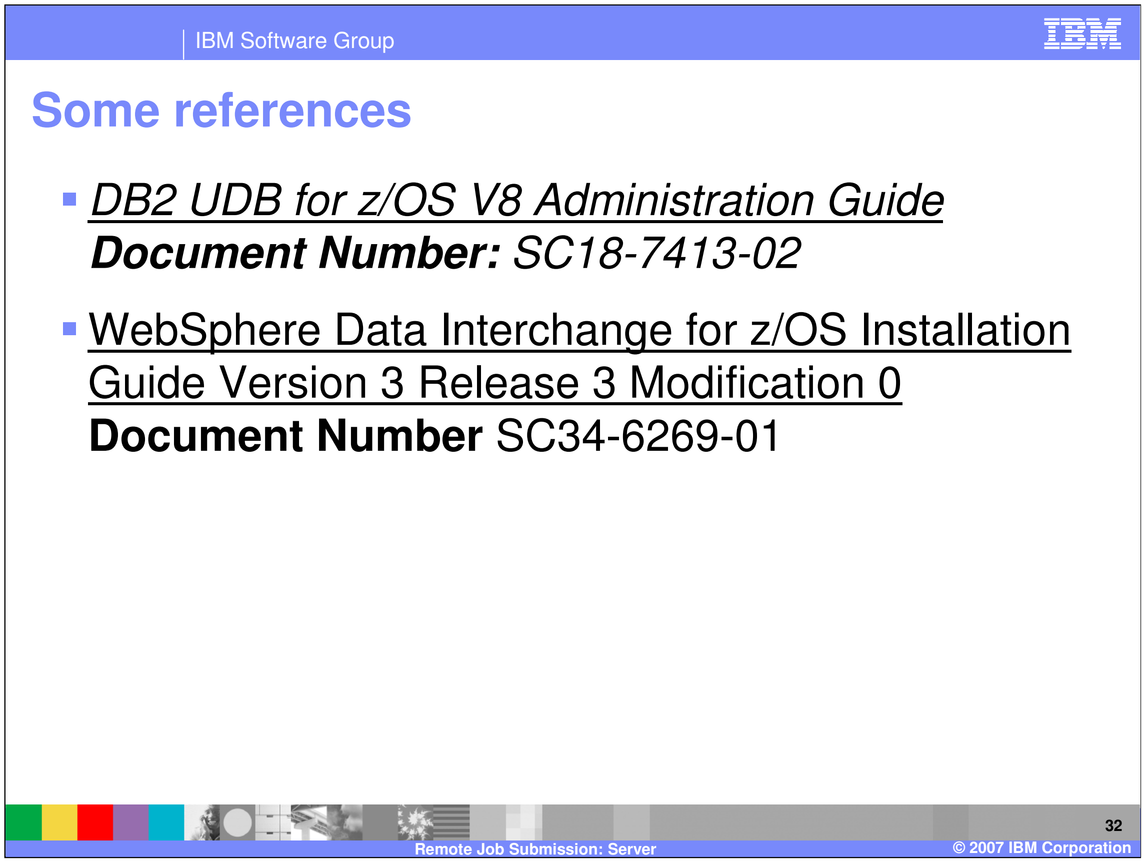


Remote Job Submission: Server

© 2007 IBM Corporation

31

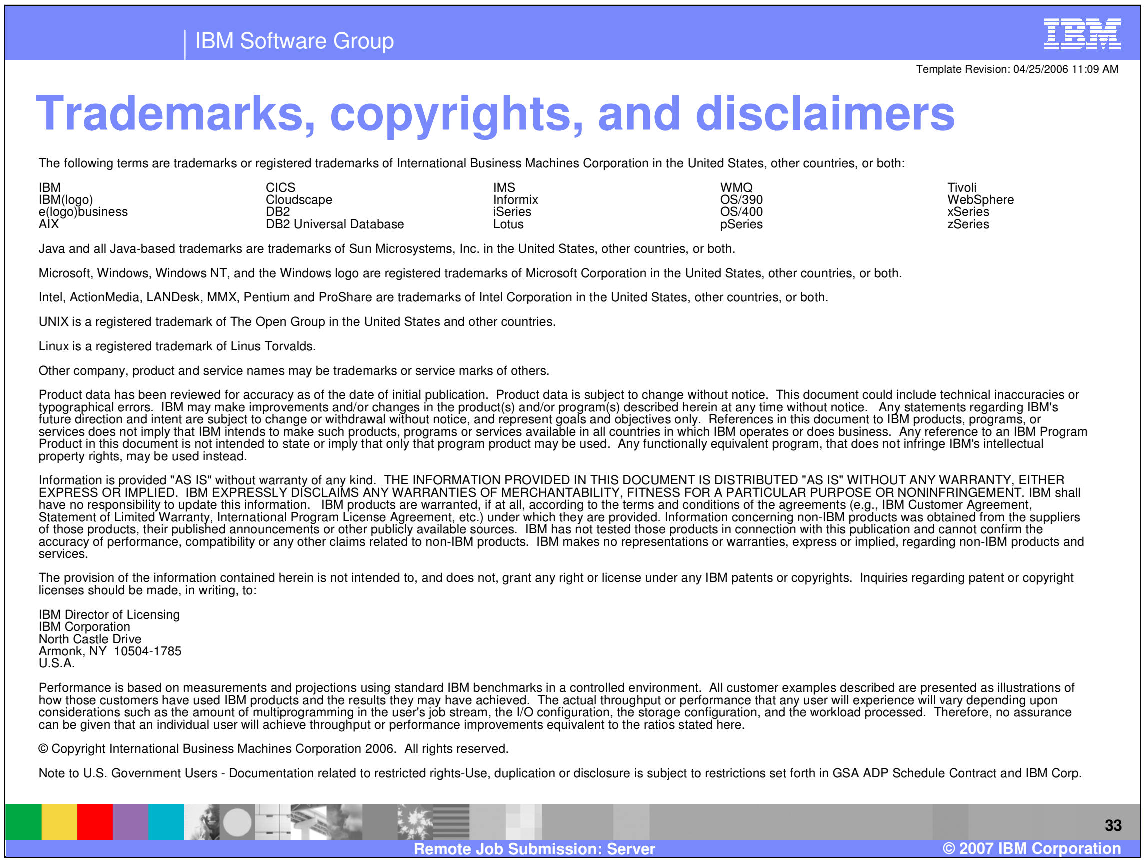
This diagram captures the execution of any request. It emphasizes the execution of processes at a remote node within an environment owned by DB2.



## Some references

- *DB2 UDB for z/OS V8 Administration Guide*
  ***Document Number:*** *SC18-7413-02*
- WebSphere Data Interchange for z/OS Installation Guide Version 3 Release 3 Modification 0
  **Document Number** SC34-6269-01

In addition to WebSphere® Data Interchange documentation, administrators need to be familiar with DB2 configuration. This is particularly true on z/OS where the Administration Guide covers many considerations directly related to DB2 and WLM. The features of the Workload Manager are covered in greater detail in other documents, and these should be considered carefully if performance and volume issues are a factor in using remote execution features.



Template Revision: 04/25/2006 11:09 AM

## Trademarks, copyrights, and disclaimers