

# 配置在ISE 2.1的ODBC与PostgreSQL

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[配置](#)

[步骤1. PostgreSQL基本配置](#)

[步骤2. ISE配置](#)

[步骤3.配置用户认证](#)

[步骤4.配置组检索](#)

[步骤5.配置属性检索](#)

[验证](#)

[故障排除](#)

[参考](#)

## 简介

本文描述如何配置身份服务引擎(ISE)使用开放数据库连接(ODBC)，用ISE验证的PostgreSQL服务器。

---

**注意：**开放数据库连接(ODBC)验证要求ISE能拿来纯文本用户密码。密码在数据库加密，但是必须由存储过程解密。

---

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- 思科身份服务引擎2.1
- 数据库和ODBC概念
- PostgreSQL

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- 身份服务引擎2.1
- Centos 7
- PostgreSQL 9.2

# 配置

**注意：**款待在本文的SQL代码为例。通常超过一种方式编码希望的功能和所有有他们的优点和缺点有。

## 步骤1. PostgreSQL基本配置

配置步骤包括数据库建立一个用户ISE的与权限访问该数据库。

### 1. 从postgres用户请创建isedb用户：

```
$ createuser --interactive
Enter name of role to add: isedb
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) y
Shall the new role be allowed to create more new roles? (y/n) n
Password:
```

### 2. 创建数据库

```
$ createdb isedb
```

或者与SQL：

```
CREATE DATABASE isedb WITH TEMPLATE = template0 OWNER = isedb;
REVOKE ALL ON DATABASE isedb FROM PUBLIC;
REVOKE ALL ON DATABASE isedb FROM postgres;
GRANT CONNECT,TEMPORARY ON DATABASE isedb TO PUBLIC;
GRANT ALL ON DATABASE isedb TO isedb;
```

### 3. 对PostgreSQL的允许

```
sudo vi /var/lib/pgsql/data/pg_hba.conf
```

在文件的底部附近查找线路看上去象的这个，：

```
host all all 127.0.0.1/32 ident
host all all ::1/128 ident
```

然后用md5替换ident，因此他们如下所示：

```
host all all 127.0.0.1/32 md5
host all all 10.0.0.0/8 md5
```

### 4. 允许对PgSQL的远程连接

您需要打开PostgreSQL配置文件/var/lib/pgsql/data/postgresql.conf。读的查找配置行：

```
listen_addresses='localhost'
```

并且更改

```
listen_addresses='*'
```

允许从所有地址的连接。注释删除端口配置线路(如果评论)：

```
port = 5432
```

### 5. 重新启动PgSQL：

```
$ sudo systemctl start postgresql
$ sudo systemctl enable postgresql
```

## 步骤2. ISE配置

创建ODBC标识来源在Administration >外部标识来源> ODBC和测试连接：

[ODBC List > pgSQL](#)

### ODBC Identity Source

General **Connection** Stored Procedures Attributes Groups

**ODBC DB connection details**

\* Hostname/IP[:port]

\* Database name

Admin username  ⓘ

Admin password

\* Timeout

\* Retries

\* Database type

**Test connection** X

Connection succeeded

**Stored Procedures**

- Plain text password authentication - Not Configured
- Plain text password fetching - Not Configured
- Check username or machine exists - Not Configured
- Fetch groups - Not Configured
- Fetch attributes - Not Configured

## 步骤3.配置用户认证

对ODBC用途存储过程的ISE验证。选择步骤的类型是可能的。在本例中我们使用参数作为返回。对于其他步骤，参考[思科身份服务引擎2.1管理指南](#)

---

**提示：**返回指定参数而不是resultset是可能的。它是不同种输出，功能是相同的。

---

### 1. 创建表。确保您设在主键的标识设置

```
CREATE TABLE "ISE_Users" (  
  user_id uuid NOT NULL,  
  username character varying NOT NULL,  
  password character varying NOT NULL  
);
```

```
ALTER TABLE public."ISE_Users" OWNER TO isedb;  
ALTER TABLE ONLY "ISE_Users"  
ADD CONSTRAINT "ISE_Users_pkey" PRIMARY KEY (user_id);
```

## 2. 运行此查询插入一个用户

```
INSERT INTO "ISE_Users" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'user1', 'password1');  
或者
```

```
INSERT INTO "ISE_Users" VALUES (uuid_generate_v1(), 'user1', 'password1');
```

并且请学习并且存储一个新用户的生成的UUID有此查询的

```
SELECT user_id FROM "ISE_Users" WHERE username = 'user1';
```

## 3. 创建纯文本密码验证的一个步骤(用于PAP , EAP-GTC内在方法 , TACACS)

```
CREATE FUNCTION iseauthuserplainreturnsparameters(ise_username text, ise_password text, OUT  
result integer, OUT ise_group text, OUT acctinfo text, OUT errorstring text) RETURNS record  
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
```

```
AS $$
```

```
DECLARE
```

```
c int;
```

```
BEGIN
```

```
select count(*) into c from "ISE_Users" where username = ise_username and password =  
ise_password;
```

```
IF c > 0 THEN
```

```
result := 0;
```

```
ise_group := cast ('11' as text);
```

```
acctinfo := cast ('This is a very good user, give him all access' as text);
```

```
errorstring := cast ('No error' as text);
```

```
else
```

```
result := 3;
```

```
ise_group := cast ('11' as text);
```

```
acctinfo := cast ('User is unknown or invalid password' as text);
```

```
errorstring := cast ('User is unknown or invalid password' as text);
```

```
END IF;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.iseauthuserplainreturnsparameters(ise_username text, ise_password text,  
OUT result integer, OUT ise_group text, OUT acctinfo text, OUT errorstring text) OWNER TO isedb;
```

## 4. 创建纯文本密码拿来的一个步骤(用于CHAP , MSCHAPv1/v2 , EAP-MD5 , LEAP , EAP-MSCHAPv2内在方法 , TACACS)

```
CREATE FUNCTION isefetchpasswordreturnsparameters(ise_username text, OUT result integer, OUT  
ise_group text, OUT acctinfo text, OUT errorstring text, OUT ise_password text) RETURNS record  
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
```

```
AS $$
```

```
DECLARE
```

```
c int;
```

```
BEGIN
```

```
select count(*) into c from "ISE_Users" where username = ise_username;
```

```
IF c > 0 THEN
```

```
result := 0;
```

```
ise_group := cast ('11' as text);
```

```
acctinfo := cast ('This is a very good user, give him all access' as text);
```

```
errorstring := cast ('no error' as text);
```

```
select password into ise_password from "ISE_Users" where username = ise_username;
```

```
else
```

```
result := 3;
```

```
ise_group := cast ('11' as text);
```

```
acctinfo := cast ('User is unknown' as text);
```

```
errorstring := cast ('User is unknown' as text);
```

```
END IF;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.isefetchpasswordreturnsparements(ise_username text, OUT result integer,
OUT ise_group text, OUT acctinfo text, OUT errorstring text, OUT ise_password text) OWNER TO
isedb;
```

## 5. 创建检查用户名的一个步骤或计算机存在(使用MAB , 快速请重新连接PEAP、EAP-FAST和EAP-TTLS)

```
CREATE FUNCTION iseuserlookupreturnsparements(ise_username text, OUT result integer, OUT
ise_group text, OUT acctinfo text, OUT errorstring text) RETURNS record
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
BEGIN
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
result := 0;
ise_group := cast ('11' as text);
acctinfo := cast ('good user' as text);
errorstring := cast ('no error' as text);
else
result := 3;
ise_group := cast ('11' as text);
acctinfo := cast ('bad user' as text);
errorstring := cast ('bad password' as text);
END IF;
END;
$$;
```

```
ALTER FUNCTION public.iseuserlookupreturnsparements(ise_username text, OUT result integer, OUT
ise_group text, OUT acctinfo text, OUT errorstring text) OWNER TO isedb;
```

## 6. 配置在ISE的步骤并且保存

[ODBC List > pgSQL](#)

### ODBC Identity Source

General	Connection	Stored Procedures	Attributes	Groups
Stored procedure type		Returns parameters		
Plain text password authentication	iseauthuserplainreturnsparements		i	+
Plain text password fetching	isefetchpasswordreturnsparements		i	+
Check username or machine exists	iseuserlookupreturnsparements		i	+
Fetch groups			i	+
Fetch attributes			i	+
Search for MAC Address in format		XX:XX:XX:XX:XX:XX	i	

## 7. 创建简单验证规则使用ODBC并且测试它

## Authentication Policy

Define the Authentication Policy by selecting the protocols that ISE should use to communicate with the network devices, and the identity sources that it should use for authentication. For Policy Export go to [Administration > System > Backup & Restore > Policy Export Page](#)

Policy Type  Simple  Rule-Based

<input checked="" type="checkbox"/>	MAB	: If Wired_MAB OR
Wireless_MAB Allow Protocols : Default Network Access and		
<input checked="" type="checkbox"/>	Default	: use Internal Endpoints
<input checked="" type="checkbox"/>	Dot1X	: If Wired_802.1X OR
Wireless_802.1X Allow Protocols : Default Network Access and		
<input checked="" type="checkbox"/>	Default	: use All_User_ID_Stores
<input checked="" type="checkbox"/>	test_aaa	: If Radius-Service-Type EQUALS Login Allow Protocols : Default Network Access and
<input checked="" type="checkbox"/>	Default	: use pgSQL
<input checked="" type="checkbox"/>	Default Rule (if no match)	: Allow Protocols : Default Network Access and use : All_User_ID_Stores

```
BAHAMUT#test aaa group ISE user1 password1 legacy
Attempting authentication test to server-group ISE using radius
User was successfully authenticated.
```

Overview	
Event	5200 Authentication succeeded
Username	user1
Endpoint Id	
Endpoint Profile	
Authentication Policy	Default >> test_aaa >> Default
Authorization Policy	Default >> Basic_Authenticated_Access
Authorization Result	PermitAccess

Authentication Details	
Source Timestamp	2016-08-26 14:18:28.17
Received Timestamp	2016-08-26 14:18:28.206
Policy Server	vumov-ise21
Event	5200 Authentication succeeded
Username	user1
Authentication Identity Store	pgSQL
Authentication Method	PAP_ASCII
Authentication Protocol	PAP_ASCII

### Steps

```
11001 Received RADIUS Access-Request
11017 RADIUS created a new session
11117 Generated a new session ID for a 3rd party NAD
15049 Evaluating Policy Group
15008 Evaluating Service Selection Policy
15048 Queried PIP - Normalised Radius RadiusFlowType (2 times)
15048 Queried PIP - Radius Service-Type
15048 Queried PIP - Normalised Radius RadiusFlowType (2 times)
15004 Matched rule - test_aaa
15041 Evaluating Identity Policy
15006 Matched Default Rule
15013 Selected Identity Source - pgSQL
24852 Perform plain text password authentication in external ODBC database - pgSQL
24849 Connecting to external ODBC database - pgSQL
24850 Successfully connected to external ODBC database - pgSQL
24856 Expect external ODBC database stored procedure to return results in output parameters - pgSQL
22037 Authentication Passed
15036 Evaluating Authorization Policy
15048 Queried PIP - Normalised Radius RadiusFlowType (4 times)
15048 Queried PIP - EndPoints LogicalProfile
15048 Queried PIP - Network Access AuthenticationStatus
15004 Matched rule - Basic_Authenticated_Access
15016 Selected Authorization Profile - PermitAccess
11002 Returned RADIUS Access-Accept
```

## 步骤4.配置组检索

### 1. 创建用于越来越多的映射的表包含用户组的和别的

```
CREATE TABLE "Groups" (
group_id uuid NOT NULL,
group_name character varying(255) NOT NULL,
group_description text
);
```

```
ALTER TABLE public."Groups" OWNER TO isedb;
```

```
ALTER TABLE ONLY "Groups"
ADD CONSTRAINT "Groups_pkey" PRIMARY KEY (group_id);
```

```
CREATE TABLE "User_Groups_Mapping" (
user_id uuid,
```

```
group_id uuid
);
```

```
ALTER TABLE public."User_Groups_Mapping" OWNER TO isedb;
```

```
ALTER TABLE ONLY "User_Groups_Mapping"
ADD CONSTRAINT "User_Groups_Mapping_group_id_fkey" FOREIGN KEY (group_id) REFERENCES
"Groups"(group_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY "User_Groups_Mapping"
ADD CONSTRAINT "User_Groups_Mapping_user_id_fkey" FOREIGN KEY (user_id) REFERENCES
"ISE_Users"(user_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

## 2. 添加组和映射，因此user1属于两组

```
INSERT INTO "Groups" VALUES ('f7dfec5c-bd06-4703-9de0-4d334ea5ec02', 'Admins', 'Group for
administrators');
INSERT INTO "Groups" VALUES ('51fc0ccd-caf8-4585-ba20-6596948c879d', 'Users', 'Group for
users');
INSERT INTO "Groups" VALUES ('7b7e72bc-ea22-470c-8578-1dd86b1a1843', 'Laptops', 'Group for users
with laptops');
```

```
INSERT INTO "User_Groups_Mapping" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'f7dfec5c-
bd06-4703-9de0-4d334ea5ec02');
INSERT INTO "User_Groups_Mapping" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', '7b7e72bc-
ea22-470c-8578-1dd86b1a1843');
```

或者请生成新的UUID，然而您将需要学习他们与挑选查询。

## 3. 创建回归类型和组检索步骤

```
CREATE TYPE g4type AS (
result integer,
group_n text
);
```

```
ALTER TYPE public.g4type OWNER TO isedb;
```

```
CREATE FUNCTION isegroupsh(ise_username text) RETURNS SETOF g4type
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
i int;
r g4type%rowtype;
BEGIN
if ise_username = '*' then
for r in select 0, cast(group_name as text) from "Groups"
loop
return next r;
end loop;
else
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
for r in select 0, cast(group_name as text) from "Groups" where group_id in (
select group_ID from "User_Groups_Mapping" where "User_Groups_Mapping".user_id IN (
select user_id from "ISE_Users" where username = ise_username
) )
loop
return next r;
end loop;
else
return query select 1,cast ('' as text);
END IF;
```

```
end if;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.isegroupsh(ise_username text) OWNER TO isedb;
```

#### 4. 映射它拿来组

[ODBC List > pgSQL](#)

### ODBC Identity Source

General    Connection    **Stored Procedures**    Attributes    Groups

Stored procedure type: Returns parameters

Plain text password authentication: iseauthuserplainreturnsparements

Plain text password fetching: isefetchpasswordreturnsparements

Check username or machine exists: iseuserlookupreturnsparements

---

Fetch groups: isegroupsh

Fetch attributes: iseattrsh

Search for MAC Address in format: XX:XX:XX:XX:XX:XX

#### 5. 拿来组并且添加他们到ODBC标识来源

[ODBC List > pgSQL](#)

### ODBC Identity Source

General    Connection    Stored Procedures    Attributes    **Groups**

Edit    + Add    X Delete

Name	Name in ISE
No data available	

Select Groups from ODBC

Sample User or Machine: \*    Retrieve Groups

<input checked="" type="checkbox"/>	Name	Name in ISE
<input checked="" type="checkbox"/>	Admins	Admins
<input checked="" type="checkbox"/>	Users	Users
<input checked="" type="checkbox"/>	Laptops	Laptops

OK    Cancel



## 6. 添加不属于任何组的另一个用户

```
INSERT INTO "ISE_Users" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'user2', 'password2');
```

## 7. 创建测验授权策略并且测试它

<input checked="" type="checkbox"/>	ODBC check Group	if	pgSQL.ExternalGroups EQUALS Admins	then	PermitAccess
<input checked="" type="checkbox"/>	Default	if no matches, then			DenyAccess

```
BAHAMUT#test aaa group ISE user1 password1 legacy
Attempting authentication test to server-group ISE using radius
User was successfully authenticated.
```

```
BAHAMUT#test aaa group ISE user2 password2 legacy
Attempting authentication test to server-group ISE using radius
User authentication request was rejected by server.
```

<b>SelectedAuthenticationIdentityStores</b>	pgSQL
<b>AuthorizationPolicyMatchedRule</b>	ODBC check Group
<b>CPMSessionID</b>	0a301a321uM9iabemtwC3JmOxm0PEPNRCy44aEudtrNg2ajmJGg
<b>ISEPolicySetName</b>	Default
<b>AllowedProtocolMatchedRule</b>	test_aaa
<b>IdentitySelectionMatchedRule</b>	Default
<b>Network Device Profile</b>	Cisco
<b>Location</b>	Location#All Locations
<b>Device Type</b>	Device Type#All Device Types
<b>ExternalGroups</b>	Admins
<b>ExternalGroups</b>	Laptops
<b>RADIUS Username</b>	user1

## 步骤5.配置属性检索

### 1. 为了简化此示例，一个平面的表使用属性

```
CREATE TABLE "User_Attributes" (
user_id uuid,
attribute_name character varying(255),
attribute_value character varying(255)
);
```

```
ALTER TABLE public."User_Attributes" OWNER TO isedb;
```

```
ALTER TABLE ONLY "User_Attributes"
ADD CONSTRAINT "User_Attributes_user_id_fkey" FOREIGN KEY (user_id) REFERENCES
```

```
"ISE_Users"(user_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

## 2. 创建两个的一个属性用户

```
INSERT INTO "User_Attributes" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'SecurityLevel', '10');
```

```
INSERT INTO "User_Attributes" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'SecurityLevel', '5');
```

```
INSERT INTO "User_Attributes" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'IdleTimeout', '5');
```

## 3. 创建回归类型和存储过程

```
CREATE TYPE a4type AS (  
result integer,  
attr_name text,  
attr_value text  
);
```

```
ALTER TYPE public.a4type OWNER TO isedb;
```

```
CREATE FUNCTION iseattrsh(ise_username text) RETURNS SETOF a4type
```

```
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
```

```
AS $$
```

```
DECLARE
```

```
c int;
```

```
r a4type%rowtype;
```

```
BEGIN
```

```
select count(*) into c from "ISE_Users" where username = ise_username;
```

```
IF c > 0 THEN
```

```
for r in select 0, cast(s.attribute_name as text), cast(s.attribute_value as text) from  
"User_Attributes" as s where user_id in(SELECT user_id from "ISE_Users" where username =  
ise_username)
```

```
loop
```

```
return next r;
```

```
end loop;
```

```
else
```

```
return query select 1, cast ('' as text);
```

```
END IF;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.iseattrsh(ise_username text) OWNER TO isedb;
```

## 4. 映射它拿来属性

## ODBC Identity Source

General    Connection    **Stored Procedures**    Attributes    Groups

Stored procedure type: Returns parameters

Plain text password authentication: iseauthuserplainreturnsparemeters

Plain text password fetching: isefetchpasswordreturnsparemeters

Check username or machine exists: iseuserlookupreturnsparemeters

---

Fetch groups: isegroupsh

Fetch attributes: iseattrsh

Search for MAC Address in format: XX:XX:XX:XX:XX:XX

### 5. 拿来属性

## ODBC Identity Source

General    Connection    Stored Procedures    **Attributes**    Groups

Edit    + Add    X Delete

Name	Type	Default Value	Name in ISE
No data available			

**Select Attributes from ODBC**

Sample User or Machine: user2    Retrieve Attributes

<input checked="" type="checkbox"/>	Name	Type	Default Value	Name in ISE
<input checked="" type="checkbox"/>	SecurityLevel	STRING	5	SecurityLevel
<input checked="" type="checkbox"/>	IdleTimeout	STRING	5	IdleTimeout

OK    Cancel

### 6. 调节ISE策略并且测试它

<input checked="" type="checkbox"/>	ODBC all access	if (pgSQL:ExternalGroups EQUALS Admins AND pgSQL:SecurityLevel EQUALS 10 )	then PermitAccess
<input checked="" type="checkbox"/>	ODBC security 5	if pgSQL:SecurityLevel EQUALS 5	then Sec-5

Status	Details	Repeat ...	Identity	End...	Endp...	Authenticati...	Authorization Policy	Authorizati...	IP
<input type="checkbox"/>			Identity	Endpc	Endpoi	Authentication	Authorization Policy	Authorization F	IP
<input checked="" type="checkbox"/>			user2			Default >> te...	Default >> ODBC security 5	Sec-5	
<input checked="" type="checkbox"/>			user1			Default >> te...	Default >> ODBC all access	PermitAccess	

## 验证

您应该当前能利用ODBC验证用户和获取他们的组和属性。

示例：

Overview	
Event	5200 Authentication succeeded
Username	user1
Endpoint ID	
Endpoint Profile	
Authentication Policy	Default == Not_Just == Default
Authorization Policy	Default == ODBC all access
Authoritative Result	PermAccess

Authentication Details	
Source Timestamp	2016-08-28 13:37:43.957
Received Timestamp	2016-08-28 13:37:43.958
Policy Server	vlmwr-0a21
Event	5200 Authentication succeeded
Username	user1
Authentication Identity Store	pgSQL
Authentication Method	PAP_PDCB
Authentication Protocol	PAP_PDCB
Service Type	Login
Network Device	Infanet
Device Type	All Device Types
Location	All Locations
NAS IPv4 Address	10.42.44.114
NAS Port Type	Async
Authoritative Profile	PermAccess
Response Time	148

Other Attributes	
ConfigVersion	103
DestinationPort	1812
Protocol	Radius
NetworkDeviceProfileName	Cisco
NetworkDeviceProfileID	403ea8b7-7a27-47c3-b096-27964031a09d
IsThirdPartyDevice	false
ActSessionID	vlmwr-0a210570121913812
SelectedAuthenticationIdentityStores	pgSQL
AuthorizationPolicyMatchedRule	ODBC all access
CPM SessionID	9a301a230-g048GwqgLF72fzV90hefwgKQu0EM0g
VSE PolicySetName	Default
AllowedProtocolMatchedRule	Not_Just
IdentitySelectorMatchedRule	Default
Network Device Profile	Cisco
Location	Location# All Locations
Device Type	Device Type# All Device Types
ExternalGroups	Admin
ExternalGroups	Laptop
SecurityLevel	10
RADIUS Username	user1

## 故障排除

如果连接不是成功的在ISE使用show logging命令应用程序prtt-management.log尾标，当尝试接通时。

错误的凭证示例：

```
2016-08-28 13:55:47,017 WARN [admin-http-pool1372][] cisco.cpm.odbcidstore.impl.PostgresDbAccess
-:admin::- Connection to ODBC DB failed. Exception: org.postgresql.util.PSQLException: FATAL:
password authentication failed for u
```

```

ser "isedb_wrong"
org.postgresql.util.PSQLException: FATAL: password authentication failed for user "isedb_wrong"
at org.postgresql.Driver$ConnectThread.getResult(Driver.java:365)
at org.postgresql.Driver.connect(Driver.java:288)
at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:208)
at com.cisco.cpm.odbcidstore.impl.PostgresDbAccess.connect(PostgresDbAccess.java:46)
at com.cisco.cpm.odbcidstore.impl.OdbcConnection.connect(OdbcConnection.java:72)
at com.cisco.cpm.odbcidstore.impl.OdbcIdStore.performTest(OdbcIdStore.java:377)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStore.testConnectionAndConfiguration(OdbcIdStore.java:469)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStoreManager.testConnectionAndConfiguration(OdbcIdStoreManager.java:84)
at com.cisco.cpm.admin.ac.actions.ODBCLPInputAction.testConnection(ODBCLPInputAction.java:749)

```

**错误的DB名称示例：**

```

2016-08-28 13:53:43,174 WARN [admin-http-pool1372][] cisco.cpm.odbcidstore.impl.PostgresDbAccess
-:admin:- Connection to ODBC DB failed. Exception: org.postgresql.util.PSQLException: FATAL:
database "isedb_wrong" does not exist
t

```

```

org.postgresql.util.PSQLException: FATAL: database "isedb_wrong" does not exist
at org.postgresql.Driver$ConnectThread.getResult(Driver.java:365)
at org.postgresql.Driver.connect(Driver.java:288)
at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:208)
at com.cisco.cpm.odbcidstore.impl.PostgresDbAccess.connect(PostgresDbAccess.java:46)
at com.cisco.cpm.odbcidstore.impl.OdbcConnection.connect(OdbcConnection.java:72)
at com.cisco.cpm.odbcidstore.impl.OdbcIdStore.performTest(OdbcIdStore.java:377)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStore.testConnectionAndConfiguration(OdbcIdStore.java:469)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStoreManager.testConnectionAndConfiguration(OdbcIdStoreManager.java:84)
at com.cisco.cpm.admin.ac.actions.ODBCLPInputAction.testConnection(ODBCLPInputAction.java:749)

```

**为了排除故障DB操作，启用日志组件odbc id存储到调试级别在管理>System >记录日志>调试日志配置下。**

**日志在prrt-management.log文件安置。**

**user1的示例：**

```

2016-08-28 14:01:01,116 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Authenticate Plain Text Password. Username=user1,
SessionID=0a301a32OuqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g
2016-08-28 14:01:01,118 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-
Write customer log message: 24852
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Authenticate plain text password
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=iseauthuserplainreturnsparameters
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Using output parameters to obtain stored procedure result values

```

2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-  
Write customer log message: 24856  
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Text: {call iseauthuserplainreturnsparameters(?, ?, ?, ?, ?, ?)}  
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Setup stored procedure input parameters, username=user1, password=\*\*\*  
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Setup stored procedure output parameters  
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Execute stored procedure call  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Process stored procedure results  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Obtain stored procedure results from output parameters  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Results successfully parsed from output parameters  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -  
:::- OdbcConnectionPool - release connection  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -  
:::- OdbcConnectionPool - connections in use: 0  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call  
to ODBC DB succeeded  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcAuthResult -:::-  
Authentication result: code=0, Connection succeeded=false, odbcDbErrorString=No error,  
odbcStoredProcedureCustomerErrorString=null, ac  
countInfo=This is a very good user, give him all access, group=11  
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-  
Write customer log message: 24853  
2016-08-28 14:01:01,129 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC  
ID Store Operation: **Get all user groups**. Username=user1,  
SessionID=0a301a320uqzqokTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g  
2016-08-28 14:01:01,131 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC  
ID Store Operation: **Fetch user groups**. Username=user1,  
SessionID=0a301a320uqzqokTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g  
2016-08-28 14:01:01,131 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write  
customer log message: 24869  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -  
:::- OdbcConnectionPool - get connection  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -  
:::- OdbcConnectionPool - use existing connection  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -  
:::- OdbcConnectionPool - connections in use: 1  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Fetch user groups  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Prepare stored procedure call, procname=**isegroupsh**  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Text: {call isegroupsh(?)}  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Setup stored procedure input parameters, username=user1  
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Execute stored procedure call  
2016-08-28 14:01:01,134 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Process stored procedure results  
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Received result recordset, total number of columns=2  
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
POSTGRES case, first column holds the result param value  
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
According to column number expect multiple rows (vertical attributes/groups returned result)  
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Fetched data: **ExternalGroup=Admins**  
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-  
Fetched data: **ExternalGroup=Laptops**

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Results successfully parsed from recordset

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Result code indicates success

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - release connection

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - connections in use: 0

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call to ODBC DB succeeded

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write customer log message: 24870

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Got groups...

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Got groups(0) = Admins

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Setting Internal groups(0) = Admins

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Got groups(1) = Laptops

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Setting Internal groups(1) = Laptops

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user groups. Username=user1, ExternalGroups=[Admins, Laptops]

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: **Fetch user attributes**. Username=user1, SessionID=0a301a320uqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write customer log message: 24872

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - get connection

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - use existing connection

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - connections in use: 1

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Fetch user attributes

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Prepare stored procedure call, procname=**iseattrsh**

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Text: {call iseattrsh(?)}

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Setup stored procedure input parameters, username=user1

2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Execute stored procedure call

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Process stored procedure results

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Received result recordset, total number of columns=3

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- **POSTGRES case**, first column holds the result param value

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- According to column number expect multiple rows (vertical attributes/groups returned result)

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Fetched data: SecurityLevel=10

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Results successfully parsed from recordset

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::- Result code indicates success

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - release connection

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -:::- OdbcConnectionPool - connections in use: 0

2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call



to ODBC DB succeeded

```
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write customer log message: 24873
```

```
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user attrs. Username=user1, Setting pgSQL.SecurityLevel to 10
```

```
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user attrs. Username=user1, Setting IdleTimeout to default value : 5
```

```
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC ID Store Operation: Get all user attrs. Username=user1, Setting pgSQL.IdleTimeout to 5
```

## 参考

- [思科身份服务引擎2.1管理指南- ODBC配置](#)
- [使用ODBC，配置与MS SQL的ISE 2.1](#)
- [PostgreSQL : 文档](#)