CFG_OIBRW Configuration

Introduction

CFG_OIBRW is another configuration record which is stored in the SYSENV system table. Its purpose is to control how the Banded Report Writer (BRW) connects to OpenInsight (i.e., OpenEngine) and which stored procedure to use as a filter for locating data sources.

Like other configuration records, CFG_OIBRW can be named to be applicable to all applications and users, or limited to a specific application and a specific user. Multiple CFG_OIBRW records can exist in this way, where the more specific configuration will always take precedence over the more generic configurations:

SYSENV Key ID	Scope
CFG_OIBRW	All applications and all users
CFG_OIBRW**AppName*	All users in the *AppName* application
CFG_OIBRW***UserName*	Only users matching *UserName*, but for all applications
CFG_OIBRW**AppName***UserName*	Only user matching *UserName* and only the *AppName* application

Default Configuration

Each attribute setting within the configuration record will have its own way of resolving default behavior (see the Settings section below for more information), but if no applicable CFG_OIBRW configuration record can be found, then helper routines like RTI_BRW_GenerateReport will assume the following settings:

```
<1> .
<3> *
<4> 0
<5> 0
<6> RTI_BRW_FILTER
```

Attributes 2, 7, 8, and 9 will be empty.

Settings

The configuration settings for CFG_OIBRW are varied and can cause unexpected behavior if not set up properly. This can be anywhere from the wrong database tables being used versus OpenInsight crashing. Therefore, while the settings information is provided for thoroughness, see the following Recommended Configurations section for examples that will work well.

Attribute	Value	Description
Connect Mode **<1>**	.	**Direct Connect** mode - The period character tells the BRW to directly connect to a local engine (via REVCAP32.DLL as opposed to the EngineServer) from the working directory (which is derived using the Drive() function). The actual engine used (i.e., in-process, existing named, or newly launched) is determined by attributes 3 and 4 (see below). This is the default value if CFG_OIBRW is missing and the default behavior if this attribute is left empty. ***Note: This applies only when using* RTI_BRW_GenerateReport*. RTI_BRWSupport's* DISPLAY *action and* Set_Printer's LOADREPORT *message both use OIPI.Net, and thus operate under a different configuration.***
	Windows Path	**Direct Connect** mode - Tells the BRW to directly connect to a local engine (via REVCAP32.DLL as opposed to the EngineServer) using the specified path. ***Note: Specifying a path other than the working directory is meant to allow the BRW to access tables local to that path. However, in practice the BRW still uses the engine from the working directory.***
	URL	**EngineServer** mode - Connects to the EngineServer at the indicated URL. Note, this could also be a local URL (e.g. **127.0.0.1)**, which works the same as **localhost** (see next entry.)
	localhost	**EngineServer** mode - Connects to the EngineServer on the local machine. This is useful for web servers running OECGI.
	!	**Special** mode - Tells the BRW to use information stored in the eserver configuration record.
	Empty	Tells the BRW to use the default setting, which is the . (i.e., period) character.
EngineServer Port **<2>**	*Port #*	**EngineServer** mode only - Port that the EngineServer is communicating through (e.g., 8088). This should be empty if attribute 1 is configured for Direct Connect mode.
	Empty	**Direct Connect** mode only - Indicates no port to be used. This is the proper value if attribute 1, *Connect Mode*, is configured for Direct Connect. This is the default value if CFG_OIBRW is missing and the default behavior if this attribute is left empty.

Engine Name **<3>**	*	The asterisk character tells the BRW to connect to the current engine (which is derived using the SYSTEM object's SERVERNAME property). This is also referred to as "sharing" the engine. This is the default value if CFG_OIBRW is missing, but it is not the default behavior if the attribute is left empty. It has a known downside of memory leaks, which can causes OpenInsight to crash when closing.
	Engine Name	The name of a specific engine, either TCP/IP Port or Named Pipe. Attribute 4, *Startup Flags*, determine if this should be an already running engine or if it should be created.
	Empty	Tells the BRW to use default behavior. As of OpenInsight v9.3, the default behavior is to create a new engine rather than share the engine.
Startup Flags **<4>**	0	Only connect to engines that already exist. Always use this value if attribute 3, *Engine Name*, has the * value. This is the default value if CFG_OIBRW is missing, but it is not the default behavior if this attribute is left empty. ***Note: Only use this value if the engine specified by attribute 3,* Engine Name*, is already running.***
	1 (or 65)	Always create a new engine. This is the default value if this attribute is left empty. 1 = Show engine UI. 65 = Hide engine UI.
	2 (or 66)	Use the current engine if possible, else create a new one. 2 = Show engine UI. 66 = Hide engine UI. ***Note: This value will use the engine specified by attribute 3,* Engine Name*, if it is already running or it will create a new engine.***
	Empty	Tells the BRW to use the default setting, which is 65 (i.e., create a new engine but hide the engine UI.)
Shutdown Flags **<5>**	0	Do not close the engine when the report is finished. Always use this value if attribute 3, *Engine Name*, has the * value. This is the default value if CFG_OIBRW is missing, but it is not the default behavior if this attribute is left empty.
	1	Close the engine when the report is finished. ***Note: Only use this setting when the Startup Flag was used to create the engine. This will have no effect on engines which already existed.***
	Empty	Tells the BRW to use the default setting, which is 1 (i.e., close the engine when the report is finished.)
Filter Routine **<6>**	*Stored Procedure*	Name of the stored procedure used by the BRW as a callback for different filter requests. The most important of these requests define which sources of data (tables, columns, or *programmatic* data sources) are allowed to be used by the BRW. This is useful if the Banded Report Designer needs to restrict access for certain users. See the CFG_OIBRW Configuration for more information. ***Note: If the filter routine is invalid the BRW will not operate.***
	Empty	Tells the BRW to use RTI_BRW_FILTER as the filter routine. This is the default value if CFG_OIBRW is missing and the default behavior if this attribute is left empty.
Unused **<7>**		
UTF8 Mode **<8>**	1	Force the BRW to use UTF8 mode.
	0	Force the BRW to use ANSI mode.
	Empty	Uses the UTF8 mode of OpenInsight (which is derived using the IsUTF8() function). This is the default value if CFG_OIBRW is missing and the default behavior if this attribute is left empty.
Debug Log **<9>**	*Log Suffix*	Tells the *RTI_BRW_GenerateReport* and *RTI_BRWSupport* routines to generate a debug log in the SYSLISTS table. The Key ID is **DEBUG_**<*Log Suffix*>. Hence, if the value in this attribute is **BRW_LOG** then the Key ID will be **DEBUG_BRW_LOG**.
	Empty	No debug log is generated.

Recommended Configurations

As noted above, not all configurations work equally well. Some will cause OpenInsight to crash or leave various processes hanging in the Task Manager. There are three basic options to consider, each one has advantages and disadvantages:

- *Direct Connect to In-Process Engine* - The advantage of this configuration is that it allows the BRW to have access to dynamically attached tables. It is also faster since it doesn't have to connect to another engine or launch one. The disadvantage of this configuration are the memory leaks which have been reported above. It does not fully support the syntax for multiple reports with a single *RTI_BRW_GenerateReport* command. Generally speaking, these leaks only manifest when OpenInsight is closed and often only when the Banded Report Designer has been used.

  Use this configuration for the **quickest implementation** setup of the Banded Report Writer:

```
<1> .
<2>
<3> *
<4> 0
<5> 0
<6> RTI_BRW_FILTER
<7>
<8>
<9>
```

- *Direct Connect to a New Engine* - The advantage of this configuration is that it is a relatively stable way to launch BRW reports. It also fully supports the syntax for multiple report usage with a single *RTI_BRW_GenerateReport* command. The disadvantage of this configuration is the lack of access to dynamically attached tables. It also requires an additional user license for the new engine that is launched, although the startup /shutdown flags can be configured to close the engine and release the user license when the report is finished.

  Use this configuration for the **most balanced** setup of the Banded Report Writer:

  ```
  <1> .
  <2>
  <3>
  <4> 65
  <5> 1
  <6> RTI_BRW_FILTER
  <7>
  <8>
  <9>
  ```

- *EngineServer Connection* - The advantage of this configuration is that it allows the BRW to offload reports to an EngineServer. This is probably the most stable configuration since all engine management is handled by the EngineServer. This also allows engines used to handling BRW requests to exist on a separate machine. The disadvantages of this configuration is the lack of access to dynamically attached tables. Additional user licenses are also needed, but these can be managed in smaller pools because of the queuing capability of the EngineServer.

  Use this configuration for the **strongest engine management** or for **special circumstances** setups of the Banded Report Writer.

  ```
  <1> <URL> (e.g., 10.0.0.15, 127.0.0.1, or localhost)
  <2> <PortNo> (e.g., 8088 or 8089)
  <3>
  <4>
  <5>
  <6> RTI_BRW_FILTER
  <7>
  <8>
  <9>
  ```