Технологический журнал

Система 1С:Предприятие 8.1 обеспечивает возможность ведения технологического журнала, в который помещается информация от всех приложений, относящихся к 1С:Предприятию 8.1.

Технологический журнал предназначен для выявления ошибок, возникающих при эксплуатации системы и диагностики работы системы службой технической поддержки фирмы «1С», а также для анализа технологических характеристик работы системы.

Состав и свойства событий технологического журнала могут меняться при выпуске обновлений платформы.

Наряду с этим, поскольку технологический журнал представляет собой набор текстовых файлов, хранящихся в различных каталогах, он может быть использован разработчиками прикладных решений для анализа различных режимов работы системы 1С:Предприятие 8.1 и прикладных решений.

Технологический журнал может вестись на любом компьютере, на котором имеется инсталляция системы 1С:Предприятие 8.1. За ведение технологического журнала отвечает конфигурационный файл, в котором описываются:

- каталог, в котором будут располагаться файлы технологического журнала;
- состав информации, которая будет помещаться в технологический журнал;
- время, в течение которого хранятся файлы технологического журнала;
- параметры дампа, создаваемого при аварийном завершении приложения.

По умолчанию конфигурационный файл отсутствует. Это означает, что технологический журнал включен, и настроен на сохранение минимальных дампов при аварийном завершении приложения в каталог C:\Documents and Settings\<ИмяПользователя>\Local Settings\Application Data\1C\1Cv8\dumps. При необходимости может быть выполнена произвольная настройка журнала регистрации с помощью отдельного конфигурационного файла. Этот файл должен иметь имя logcfg.xml и распологатья в подкаталоге conf каталога программных файлов 1С:Предприятия 8.1 (bin).

Система 1С:Предприятие 8.1 автоматически с периодичностью 60 секунд опрашивает каталог программных файлов на наличие конфигурационного файла и его состав. Таким образом, изменение параметров технологического журнала может быть выполнено «на ходу», без перезапуска работающих приложений 1С:Предприятия 8.1.

При определенных настройках объем технологического журнала может быть достаточно большим, поэтому в конфигурационном файле желательно указывать время, в течение которого хранятся файлы журнала. По истечении указанного времени система 1С:Предприятие 8.1 удалит устаревшие файлы журнала. Если после удаления устаревших файлов каталог, в котором располагались эти файлы, оказывается пустым, то такой каталог тоже удаляется. Таким образом, все дерево каталогов технологического журнала не содержит устаревших файлов и папок.

Если работа системы выполняется в среде Linux, управление выдачей аварийных дампов (core) выполняется средствами операционной системы. При этом в технологический журнал помещается информация о факте аварийного завершения процесса и о номере сигнала, повлекшего за собой это завершение.

Конфигурационный файл технологического журнала

Для настройки параметров технологического журнала предназначен конфигурационный файл. Конфигурационный файл располагается в подкаталоге conf каталога загрузочных модулей 1С:Предприятия 8.1 (bin). Он называется logcfg.xml и имеет формат XML.

В простейшем виде конфигурационный файл может иметь, например, следующее содержимое:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\logs" history="1">
        <event> <eq property="Name" value="CONN"/> </event>
    </log>
    <dump location="c:\v81\dumps" create="1" type="2"/>
</config>
```

Данный конфигурационный файл указывает, что:

- в технологическом журнале регистрируются все события, установки и разрыва клиентского соединения с сервером;
- файлы технологического журнала располагаются в каталоге c:\v81\logs;
- файлы технологического журнала хранятся в течение одного часа;
- файлы дампа помещаются в каталог c:\v81\dumps;
- файлы дампа содержат всю доступную информацию (содержимое всей памяти процесса).

При отсутствии конфигурационного файла используется следующие параметры:

- дампы минимального размера;
- Каталоги дампов создаются в каталоге C:\Documents and Settings\<ИмяПользователя>\Local Settings\Application Data\1C\1Cv81\.

Имя каждого подкаталога технологического журнала одного процесса будет иметь вид: <ИмяПроцесса>_<ИдентификаторПроцесса>, например: rphost_2488.

Структура конфигурационного файла

Корневым элементом конфигурационного файла является элемент <config>, который определяет настройки технологического журнала. Он может содержать несколько элементов <log> и один элемент <dump>.

```
<config …>
    <log …> … </log>
    <log …> … </log>
    <log …> … </log>
    <dump … />
</config>
```

Элемент <log> определяет каталог технологического журнала. Его атрибуты:

- location — имя каталога, в котором будет размещаться технологический журнал;

Необходимо иметь в виду, что каталог технологического журнала не предназначен для хранения в нем файлов, которые не относятся к технологическому журналу. Поэтому не следует размещать в нем дампы. Очистка каталога технологического журнала от устаревших файлов журнала выполняется 1 раз в час. При этом посторонние файлы, помещенные в каталог технологического журнала, удаляются.
В элементах:
<log location="каталог технологического журнала" history="24"></log>
и
<dump location="каталог дампов" create="1" type="2"/>
следует указывать разные каталоги.

- history — количество часов, через которое информация будет удаляться из технологического журнала.

В элемент <log> могут быть вложены элементы <event> и <property>, состав которых определяет условие записи в журнал каждого события и условия записи каждого свойства события.
Если этот элемент не содержит ни одного элемента <event>, то никакие события в журнал записываться не будут.
Элемент <dump> определяет каталог для записи дампов аварийного завершения.

Элемент <event>

Последовательность элементов <event> определяет условие, при выполнении которого событие будет помещено в журнал. В журнал помещаются только такие события, которые удовлетворяют условию. Иначе говоря, если условие, определяемое последовательностью элементов <event>, принимает значение «Истина», то событие будет записано в журнал. Событие включается в журнал, если оно удовлетворяет всем условиям внутри хотя бы одного из элементов <event>. То есть условия внутри <event> объединяются по «И», а элементы <event> объединяются по «ИЛИ».
Условия задаются элементами:

- eq — равно;
- ne — не равно;
- gt — больше;
- ge — больше или равно;
- lt — меньше;
- le — меньше или равно;
- like — соответствие маске.

Каждый из этих элементов, кроме элемента like определяет простое сравнение значения параметра события (имя которого задается атрибутом property) со значением атрибута value. Например:

```
<event>
    <eq property="Name" value="PROC"/>
</event>
```

В данном случае в технологическом журнале будут регистрироваться события, относящиеся к группе с именем PROC. Доступны следующие имена групп событий:

- PROC — события, относящиеся к процессу целиком, и влияющие на дальнейшую работоспособность процесса. Например: старт, завершение, аварийное завершение и т.п.;
- SCOM — события создания или удаления серверного контекста, обычно связанного с информационной базой;
- EXCP — исключительные ситуации, приложений системы 1С: Предприятие 8.1, которые штатно не обрабатываются и могут послужить причиной аварийного завершения серверного процесса или подсоединенного к нему клиентского процесса;
- EXCPCNTX — события, которые начались, но не закончились в момент возникновения нештатной ситуации;
- SDBL — события, связанные с исполнением запросов к модели базы данных 1С:Предприятия 8.1;
- QERR – события, связанные с обнаружением ошибок компиляции запроса или ограничения на уровне записей и полей базы данных;
- PERR — события, связанные с обнаружением ошибок работы с настройками пользователя;
- CONN — установка или разрыв клиентского соединения с сервером;
- ADMIN — управляющие воздействия администратора кластера серверов 1С:Предприятия 8.1;

- DBV8DBEng — исполнение операторов SQL файловой СУБД;
- DBMSSQL — исполнение операторов SQL СУБД Microsoft SQL Server;
- DBPOSTGRS — исполнение операторов SQL СУБД PostgreSQL;
- DB2 — исполнение операторов SQL СУБД DB2;
- CALL — удаленный вызов;
- TLOCK — управление транзакционными блокировками в Управляемом режиме.

Следует заметить, что на клиентском компьютере могут возникать только те события, которые относятся к группам PROC, EXCP, SDBL. Также на клиентском компьютере могут возникать события из группы DBV8DBEng, если используется файловый вариант работы системы 1С:Предприятие 8.1.
Также следует заметить, что события из групп PROC, SCOM, EXCP, CONN и ADMIN возникают относительно редко и содержат небольшое количество информации, в то время как регистрация событий из групп SDBL, DBV8DBEng и DBMSSQL может приводить к значительному росту объема технологического журнала.
Элемент like определяет, соотстветствует ли свойство события технологического журнала некоторой маске. Маска представляет собой последовательность символов, некоторые из которых означают сами себя, а некоторые являются шаблонами и служат для описания группы символов.
Например, элемент

```
<like property="sql" value="%reference%"/>
```

означает проверку значения свойства sql события технологического журнала на соответствие маске %reference%
К шаблонам относятся:

- % - 0 или более произвольных символов,
- _ - 1 произвольный символ,
- [...] - один из перечисленных символов, причем … может содержать произвольную символов, а также диапазоны вида с-С, где с – начальный символ диапазона, С – конечный символ диапазона,
- [^...] - один любой символ, кроме перечисленных,
- \ - префиксный символ, игнорируется и означает, что следующий за ним символ – это просто символ, означающий сам себя (а не шаблон).

Все другие символы – это простые символы, которые означают сами себя. При сравнении простых символов регистры букв не различаются.
Примеры шаблонов.
"шаблон" – строка с конкретным текстом. В этом случае сравнение like не отличается от сравнения eq. Регистры букв не различаются.
"%reference%" – строка, содержащая контекст reference в произвольном месте. Регистры букв не различаются.
"reference%" – строка, содержащая контекст reference в начале. Регистры букв не различаются.
"%reference" – строка, содержащая контекст reference в конце. Регистры букв не различаются.
"%[a-z]" – строка с маленькой английской буквой от a до z в конце.
"%[^a-z]%" – строка, содержащая хотя бы один символ, отличающийся от маленькой английской буквы.

Замечание. Фильтрация событий по шаблонам медленнее, чем при использовании других элементов сравнения. Использование сложной фильтрации событий и свойств технологического журнала может несколько замедлить работу 1С:Предприятия.

В каждом элементе <event> может быть задано несколько условий. В этом случае условия объединяются логической операцией «И». Если элементов <event> несколько, то если хотя бы один из них выполняется, то событие попадает в журнал. Например:

```
<log location="c:\logs" history="1">
    <event> <eq property="Name" value="PROC"/> </event>
    <event> <eq property="Name" value="SCOM"/> </event>
    <event> <eq property="Name" value="CONN"/> </event>
    <event> <eq property="Name" value="EXCP"/> </event>
    <event> <eq property="Name" value="DBMSSQL"/> </event>
</log>
```

В данном примере указывается, что в технологическом журнале будут регистрироваться события, относящиеся к группам PROC, SCOM, CONN, EXCP и DBMSSQL.
Все события имеют свойство process=<ИмяПриложения>, например,

```
30:28.9222-1,DBMSSQL,2, process=rphost,p:processName=e00074708,
t:clientID=17,t:applicationName=Фоновое задание,t:computerName=,
t:connectID=14,Sql=SELECT * FROM DBSchema,Rows=1
```

Элемент <property>

Элемент <property> определяет условия попадания в журнал значения ключевого свойства события, имя которого является значением атрибута name, при условии, что само событие в журнал попадает. Условия задаются вложенными элементами <event> по таким же правилам, что и для событий.

Если элемент <property> с определенным именем отсутствует, то соответствующее свойство не пишется. Если элемент <property> не содержит вложенных элементов <event>, то определяемое им свойство пишется для всех событий, попадающих в журнал, в которых оно присутствует. Если элемент <property> содержит вложенные элементы <event>, то свойство будет записано только для событий, удовлетворяющих условию (если само событие в журнал записывается, и событие имеет данное свойство).
Элемент <property name=”all”> </property> включает записи в журнал всех свойств событий.
Приведенный ниже элемент <log> определяет запись в журнал событий: процесса, серверного контекста, соединения, исключений и исполнение операторов SQL. Причем, текст оператора SQL будет помещен в журнал только, если он исполнялся более секунды. Журнал располагается в каталоге c:\logs и хранится 1 час.

```
<log location="c:\logs" history="1">
    <event> <eq property="Name" value="PROC"/> </event>
    <event> <eq property="Name" value="SCOM"/> </event>
    <event> <eq property="Name" value="CONN"/> </event>
    <event> <eq property="Name" value="EXCP"/> </event>
    <event> <eq property="Name" value="DBMSSQL"/> </event>
    <property name="sql">
        <event>
            <eq property="Name" value="MSSQL"/>
            <gt property="Duration" value="10000”/>
        </event>
    </property>
</log>
```

Каждое событие имеет набор свойств. Каждое свойство имеет имя. Возможно присутствие в событии нескольких ключевых свойств с одинаковыми именами. Имена свойств могут использоваться для фильтрации событий и свойств. Большие и малые буквы при сравнении имен не различаются. Пустое условие в элементе <property> будет означать, что свойство будет выводиться при любом условии.

Примечание. Свойство события выводится, только если для него присутствует элемент <property>.

Далее перечислены основные свойства событий, которые могут потребоваться для настройки конфигурационного файла или просмотра технологического журнала:

- Administrator — имя администратора кластера или центрального сервера;
- Process — наименование приложения, как его представляет операционная система (имя файла загрузочного модуля приложения);
- Calls — количество обращений клиентского приложения к серверному приложению через TCP;
- Cluster — номер основного порта кластера серверов;
- Connection — номер соединения с информационной базой;
- Context — контекст исполнения;
- Dbpid — строковое представление идентификатора соединения сервера 1С:Предприятия с сервером баз данных в терминах сервера баз данных (для событий DBMSSQL, DBPOSTGRS, DB2);
- Descr — пояснения к программному исключению;
- DumpError — описание ошибки, произошедшей в процессе построения дампа;
- DumpFile — имя файла с дампом;
- Duration — длительность события в десятитысячных долях секунды;
- Err — консольное сообщение: 0 — информационные, 1 — об ошибке;
- Exception — наименование программного исключения;
- Finish — причина завершения процесса;
- Func — наименование выполняемого действия:
    - beginTransaction — начало транзакции (событие типа SDBL выводится в журнал в момент начала транзакции и не имеет длительности);
    - ransaction — начало транзакции (событие типа SDBL начинается при начале транзакции, заканчивается – при завершении транзакции);
    - commitTransaction — фиксация транзакции;
    - rollbackTransaction — отмена транзакции;
    - setRollbackOnly — установка флага наличия в транзакции ошибки (ее можно только откатить);
    - getTransactionSplitter — получение разделителя итогов;
    - quickInsert — быстрая вставка данных в таблицу базы данных;
    - insertRecords — добавление записи в таблицу базы данных;
    - suspendIndexing — отмена индексирования таблиц базы данных;
    - resumeIndexing — восстановление индексирования таблиц базы данных;
    - holdConnection — удержание соединения;
    - saveObject — сохранение объекта;
    - restoreObject — восстановление объекта;
    - readFile — чтение файла;
    - createFile — создание файла;
    - deleteFile — удаление файла;
    - searchFile — поиск файла;
    - modifyFile — обновление файла;
    - isProperLocale — проверка национальных настроек, установленных для базы данных;
    - changeLocale — изменение национальные настройки базы данных;
    - takeKeyVal — получение значения ключа записи табличной части;
    - lockRecord — блокировка записи;

- serializeTable — сохранение данных таблицы в файл;
- deserializeTable — восстановление данных таблицы базы данных из файла;
- xlockTables — установка исключительной блокировки на таблицу;
- xlockTablesShared — установка разделяемой блокировки на таблицу;
- copyMoveFile –копирование/ перемещение фрагмента конфигурации между записями таблиц базы данных;
- moveFile — перемещение файла;
- securedInsert –вставка записей с наложением ограничений доступа к данным;
- selectFileName — выбор имени файла;
- setSingleUser — установка монопольного режима;
- insertIBRegistry — создание кластера;
- eraseIBRegistry — удаление кластера;
- setRegMultiProcEnable — установка значения флага поддержки кластером многих рабочих процессов;
- setServerProcessCapacity — установка значения пропускной способности рабочего процесса;
- agentAuthenticate –аутентификация администратора центрального сервера;
- insertAgentUser — добавление администратора центрального сервера;
- eraseAgentUser — удаление администратора центрального сервера;
- setRegSecLevel — установка уровня безопасности кластера;
- setRegDescr — установка описания кластера;
- setInfoBaseDescr — установка описания информационной базы;
- insertServerProcess — добавление рабочего процесса;
- eraseServerProcess — удаление рабочего процесса;
- regAuthenticate — аутентификация администратора кластера;
- insertRegUser — добавление администратора кластера;
- eraseRegUser — удаление администратора кластера;
- setServerProcessEnable — установка значения флага разрешения запуска рабочего процесса;
- insertRegServer — добавление рабочего сервера;
- eraseRegServer — удаление рабочего сервера;
- updateRegServer — изменение параметров рабочего сервера;
- authenticateAdmin –аутентификация администратора информационной базы;
- createInfoBase — создание информационной базы;
- dropInfoBase — удаление информационной базы;
- killClient — разрыв соединения клиента с кластером серверов 1С:Предприятия;
- authenticateSrvrUser — аутентификация администратора кластера в рабочем процессе
- setInfoBaseConnectingDeny — установка режима блокировки установки соединений с информационной базой;
- lookupTmpTable — получение/ создание временной таблицы базы данных;
- returnTmpTable — освобождение временной таблицы базы данных;

· Host — имя компьютера;
· Name — имя события;
· OLEDBException — описание исключения от Microsoft SQL Server;

- OSException — описание исключения операционной системы;
- p:processName — имя серверного контекста, который обычно совпадает с именем информационной базы;
- Port — номер основного IP порта процесса;
- ProcessName — наименование процесса;
- Reason — номер исключения;
- Ref — имя информационной базы;
- Rows — количество полученных записей базы данных;
- RowsAffected — количество измененных записей базы данных;
- RunAs — режим запуска процесса (приложение или сервис);
- SDBL — текст запроса на встроенном языке модели базы данных;
- ServerName — имя рабочего сервера;
- Sql — текст оператора SQL;
- SyncPort — номер вспомогательного IP порта процесса;
- t:applicationName — идентификатор клиентской программы;
- t:clientID — идентификатор соединения с клиентом по TCP;
- t:computerName — имя клиентского компьютера;
- t:connectID — идентификатор соединения с информационной базой;
- Trans — идентификатор активности транзакции на момент начала события (1 – транзакция была открыта, 0 — транзакция не была открыта);
- Txt — текст консольного сообщения;
- V8DBEngException — описание ошибки СУБД для файловой СУБД;

Val — значение, смысл зависит от значения параметра Func;

Используя свойства элемента <property>, в технологический журнал можно записывать контекст исполнения. Контекст исполнения может быть двух видов: контекст встроенного языка и интерфейсный контекст. Контекст встроенного языка представляет собой список операторов встроенного языка и содержит в себе:

- Название модуля;
- Номер строки модуля;
- Текстовое представление элемента списка вызова встроенного языка соответствующей строки модуля.

Интерфейсный контекст включает в себя:

- Полное имя формы;
- Тип активного элемента формы;
- Имя активного элемента формы;
- Имя кнопки командной панели (если она была нажата);

Действие, выполняемое элементом формы;

Например:

Контекст встроенного языка в файле технологического журнала может иметь вид:

```
Документ.ПриходнаяНакладная : 23 : Движения.УчетНоменклатуры.Записать();
МодульПриложения : 18 : ПроверитьПодключениеОбработчикаОжидания(Истина);
МодульПриложения : 230 : Если нпПолучитьЗначениеПоУмолчанию
(глТекущийПользователь, "ИспользоватьНапоминания")
ОбщийМодуль.нпНастройкиПользователей : 481 : Выборка =
Запрос.Выполнить().Выбрать();
```

Интерфейсный контекст в файле технологического журнала может иметь такой вид:

```
{Документ.Документ1.ФормаСписка}/{ТабличноеПоле :
            ДокументСписок}/{ОбновлениеОтображения}
{Документ.Документ1.Форма.ФормаДокумента}/{ КоманднаяПанель :
          ОсновныеДействияФормы}/{ОсновныеДействияФормыОК}
{Документ.Документ1.Форма.ФормаДокумента}/{ Кнопка :
            Кнопка1}/{Нажатие}
```

Чтобы включить запись контекста нужно среди фильтров свойств записать элемент <property name="context"> или элемент <property name="all">. Например:

Если нужно записывать события SDBL (SDBL-запросы) и DBMSSQL (операторы SQL к СУБД Microsoft SQL Server) с контекстом исполнения:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\1logs" history="1">
        <event>
            <eq property="Name" value="SDBL"/>
        </event>
        <event>
            <eq property="Name" value="DBMSSQL"/>
        </event>
        <property name="Context">
        </property>
    </log>
</config>
```

Чтобы записывать события SDBL (SDBL-запросы) и DBMSSQL (операторы SQL к СУБД Microsoft SQL Server) без контекста исполнения:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\1logs" history="1">
      <event>
            <eq property="Name" value="SDBL"/>
        </event>
        <event>
            <eq property="Name" value="DBMSSQL"/>
        </event>
    </log>
</config>
```

Чтобы записывать события SDBL (SDBL-запросы) и DBMSSQL (операторы SQL к СУБД Microsoft SQL Server) без контекста исполнения, но со всеми другими свойствами:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\1logs" history="1">
        <event>
            <eq property="Name" value="SDBL"/>
        </event>
        <event>
            <eq property="Name" value="DBMSSQL"/>
        </event>

        <property name="all">
        </property>
        <property name="Context">
           <eq property="Name" value=""/>
        </property>
    </log>
</config>
```

Для того, чтобы записывать события SDBL (SDBL-запросы) с контекстом исполнения и DBMSSQL (операторы SQL к СУБД Microsoft SQL Server) без контекста исполнения:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\1logs" history="1">
        <event>
            <eq property="Name" value="SDBL"/>
        </event>
```

```
        <event>
            <eq property="Name" value="DBMSSQL"/>
        </event>
        <property name="Context">
            <event>
                <eq property="Name" value="SDBL"/>
             </event>
        </property>
    </log>
</config>
```

Наличие элемента <property name="Context"> означает, что для записываемых в журнал событий при выполнении условий, указанных в данном элементе, будет записана информация о контексте. После этого в каждое событие технологического журнала будет добавлена информация о контексте исполнения в текущем процессе, а после события будет добавлено мгновенное событие, несущее информацию о контексте исполнения клиентского процесса.

В технологический журнал могут быть записаны сообщения об исключительных ситуациях, связанных с менеджером блокировок. Для этого файл конфигурации должен иметь примерно следующий вид:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
    <log location="c:\v81\logs" history="7">
        <event>
            <eq property="Name" value="EXCP"/>
        </event>
        <event>
            <eq property="Name" value="TLOCK"/>
            <gt property="Duration" value="100000"/>
        </event>
        <property name="all"/>
        <property name="Context">
            <event>
                <eq property="Name" value=""/>
            </event>
        </property>
    </log>
    <dump location="c:\v81\dumps" create="1" type="2"/>
</config>
```

В приведенном случае будут регистрироваться все исключительные ситуации, связанные с блокировками (в частности DEADLOCK — взаимные блокировки сессий и TIMEOUT — истечение предопределенного времени, при этом в обоих случаях в текст сообщения об исключительной ситуации включается номер сессии, которая вызвала эту исключительную ситуацию) и ожидания, превысившие 10 секунд. При этом будет записана информация по âñåм ñâîéñòâàì, êðîìå Context.

Элемент <dump>

Элемент <dump> определяет параметры дампа, создаваемого при аварийном завершении приложения. Чтобы отключить запись дампов нужно в элементе <dump> установить значение параметра create = "0" или create = "false". Если элемент <dump> отсутствует, то для записи дампов будет использоваться каталог %USERPROFILE%\Local Settings\Application Data\1C\1Cv81\Dumps.

Его атрибуты:

- location — имя каталога, в который будут помещаться файлы дампов;

- create — 0 ("false") — не создавать, 1 ("true") — создавать;
- type — тип дампа, произвольная комбинация следующих флагов, представленная в десятичной или шестнадцатеричной системе. Представление в шестнадцатеричной системе должно начинаться с символа ‘x’, например, x0002. Доступны следующие значения:
  - 0 (x0000) — минимальный;
  - 1 (x0001) — дополнительно сегмент данных;
  - 2 (x0002) — содержимое всей памяти процесса;
  - 4 (x0004) — данные хэндлов;
  - 8 (x0008) — оставить в дампе только информацию, необходимую для восстановления стеков вызовов;
  - 16 (x0010) — если стек содержит ссылки на память модулей, то добавить флаг 0x0040;
  - 32 (x0020) — включить в дамп память из-под выгруженных модулей;
  - 64 (0x0040) — включить в дамп память, на которую есть ссылки;
  - 128 (x0080) — добавить в дамп подробную информацию о файлах модулей;
  - 256 (0x0100) — добавить в дамп локальные данные потоков;
  - 512 (0x0200) — включение в дамп памяти из всего доступного виртуального адресного пространства;
- prntscrn – тип булево. Истина - создать файл копии экрана при аварийном завершении клиентской части 1С:Предприятия 8.1. Файлы копии экрана создаются в том же каталоге, что и дампы (атрибут location). Имя файла совпадает с именем файла дампа, но имеет расширение png. По умолчанию копии экрана не создаются.

При аварийном завершении программы 1С:Предприятие 8.1, система выдает диалог с информацией о процессе записи дампа, который автоматически закрывается после завершения записи дампа.

Вывод контекстов исключений

Контекст исключения представляет собой последовательность событий технологического журнала типа EXCPCNTX. Каждое событие типа EXCPCNTX является одним из длительных событий, которые начались, но не закончились в момент возникновения нештатной ситуации в работе 1С:Предприятия. При этом события выводятся в порядке убывания уровня вложенности. Тип события, исходного для события EXCPCNTX, становится значением свойства SrcName события EXCPCNTX. Контекст исключения выводится в технологический журнал, если технологический журнал включен (в файле logcfg.xml имеется хотя бы один элемент log) и произошла одна из следующих нештатных ситуаций:

- при работе 1С:Предприятия произошло исключение операционной системы, процесс 1С:Предприятия (клиент или сервер) завершен аварийно, построен дамп аварийного завершения;
- возникло исключение от базы данных, приводящее к отображению сообщения об ошибке и закрытию приложения 1С:Предприятия.

При возникновении любой ошибки базы данных в технологический журнал записывается событие типа EXCP, если оно удовлетворяет условиям, записанным в конфигурационном файле технологического журнала (logcfg.xml).

Вывод информации о взаимных блокировках

При любом обращении к СУБД, но не чаще, чем 1 раз в 2 секунды, выполняется дополнительное обращение к СУБД с запросом, какой поток был заблокирован и каким потоком. Результатом такого запроса является таблица пар («жертва блокировки», «источник блокировки»), где:

- «жертва блокировки» — идентификатор соединения с СУБД, которое ожидает блокировки;
- «источник блокировки» — идентификатор соединения с СУБД, которое установило блокировку.

Если в кластере несколько рабочих процессов, то запрос выполняется одним из них. Запросы о взаимных блокировках нумеруются.
Данные из полученной таблицы добавляются к контексту каждого потока, которым соответствуют полученные идентификаторы соединений с СУБД, и будут отображены в качестве значения блокировочных свойств очередного события технологического журнала. После того, как в потоке, к контексту которого добавлена информация о блокировках, будет завершено очередное событие технологического журнала, к этому событию будут добавлены блокировочные свойства. При этом если поток был жертвой блокировки, то события блокировки будут очищены после вывода. Если поток был источником, то очистка выполняется при закрытии или откате транзакции.
Информация о блокировках добавляется к потокам в следующем порядке:
Если поток – жертва еще не знает об этом, то ему устанавливается номер запроса и идентификатор потока – источника блокировки.
К потоку – источнику блокировки добавляется номер запроса только, если у него есть жертвы, которые об этом еще не знали.

Информация о блокировках:

- поток является источником, момент обнаружения;
- поток является жертвой, момент обнаружения;
- номер запроса (если поток является жертвой);
- список номеров запросов (если поток является источником);
- номер соединения источника (если поток является жертвой).

Блокировочные свойства событий:
lka‘1’ — Поток является источником блокировки.
lkp‘1’ — Поток является жертвой блокировки.
lkpid — Номер запроса к СУБД «кто кого заблокировал» (только для потока – жертвы блокировки). Например, ‘423’
lkaid — Список номеров запросов к СУБД «кто кого заблокировал» (только для потока – источника блокировки). Например, ‘271,273,274’
lksrc — Номер соединения источника блокировки, если поток является жертвой, например, ‘23’
lkpto — Время в секундах, прошедшее с момента обнаружения, что поток является жертвой. Например: ‘15’
lkato — Время в секундах, прошедшее с момента обнаружения, что поток является источником блокировок. Например, ‘21’
Таким образом, для анализа блокировок необходимо найти в технологических журналах процессов rphost первое событие со свойствами lka и lkp, узнать значения свойств lkaid, lkpid и найти все события с этими значениями свойств в журналах всех рабочих процессов кластера. По найденной группе событий можно установить, кто кого заблокировал, на сколько времени и что они при этом детали.

Структура технологического журнала

Технологический журнал представляет собой каталог, в подкаталогах которого располагаются файлы с собранными технологическими данными. Каталог журнала имеет следующую структуру:

<каталог журнала>

<идентификатор процесса операционной системы>

<файлы журнала одного процесса>

Каждый файл журнала содержит события за 1 час и имеет имя yymmddhh.log, где:

- yy — две последние цифры года;
- mm — номер месяца;
- dd — номер дня;
- hh — номер часа.

Файлы журнала имеют текстовый формат. В файле сведения о завершении каждого события записываются с новой строки. Например:

```
11:33.0312-1,DBV8DBEng,2,Func=selectFileName,FileName=versions.new
11:33.0313-3,DBV8DBEng,1,Func=modifyFile,CatName=ConfigSave,FileName=
versions.new,What=content
```

Строка окончания события имеет формат:

mm:ss.tttt-d, <наименование>, <уровень>, <ключевые свойства>

- mm — номер минуты в текущем часе;
- ss — номер секунды в текущей минуте;
- tttt — номер десятитысячной доли текущей секунды;
- d — длительность события в десятитысячных секунды;
- <наименование> — наименование события;
- <уровень> — уровень события в стеке текущего потока;
- <ключевые свойства> — <ключевое свойство>, <ключевое свойство>, …;
- <ключевое свойство> — <имя> = <значение>

<наименование>, <имя>, <значение> — произвольный текст. Если в нем присутствуют символы «конец строки» или «запятая», то текст заключается в кавычки или апострофы в зависимости от того, каких символов в строке меньше, а кавычки или апострофы в тексте удваиваются.