

## Генерация ежедневных отчётов об аварийных событиях в ReportWorX

Одной из часто встречающихся задач на рынке автоматизации является внедрение системы отчётов о состоянии технологических процессов. В частности, наиболее интересна для заказчиков информация о произошедших за определённый временной период аварийных событиях. Такая информация позволяет прогнозировать ресурс работы оборудования и планировать планово-предупредительные и аварийные ремонтные работы.

С подобной задачей на одном из объектов столкнулась и инжиниринговая компания «Инсайт-НГ». Необходимо было разработать систему генерации периодических (суточных) отчётов о возникших аварийных событиях в технологических системах обслуживаемого объекта с последующей рассылкой отчётов по заданным адресам электронной почты.

Для решения задачи был выбран компонент ReportWorX из пакета BizViz компании ICONICS, выбор был определён тем, что на объекте уже стояла SCADA-система GENESIS64 разработки компании ICONICS. Хотя многие компоненты BizViz 32-битовые, они прекрасно работают с GENESIS64.

ReportWorX – это система документирования, обеспечивающая создание, исполнение и перенаправление отчётов. Система работает как с продуктами ICONICS, так и с другими источниками данных: различные SCADA-системы, данные MES и ERP-систем, корпоративные базы данных и т.д. Поддерживается доступ к отчётам из Интернета и корпоративной сети.

В качестве источника данных для формирования отчётов используется база данных AlarmWorX64 Logger. Она хранит всю информацию о произошедших аварийных событиях и наполняется автоматически.

Далее поэтапно описан процесс конфигурирования ReportWorX и разработки необходимых SQL-запросов к базе. Поскольку статья не ставит своей целью подменить руководство ICONICS, а также объяснить правила разработки SQL-запросов, ряд моментов был сознательно опущен.

### Предварительные настройки

Разработка велась в следующей конфигурации системы:

- Windows 7 Professional x64;
- ICONICS GENESIS64 v.10.71, HotFixPack 1, ServicePack 1, ServicePack 2;
- BizViz 9.22, компонент ReportWorX;
- Microsoft Excel 2007.

В AlarmWorX64 Server были созданы аварийные сигналы, связанные с регистрами UDM, а также *Area* – области для группировки аварийных сигналов, например, для разных объектов, типов систем, систем. Созданные *Area* в дальнейшем использовались при формировании запросов в базу данных AlarmWorX64 Logger и были организованы по иерархическому принципу:

1) *Area* с названием АРМ, куда выводится информация по системе (АРМ Э, АРМ ОВК и т.д.);

а. *Area*-названия объектов (внешняя территория, КТП, административный корпус и т.д.);

і. *Area*-названия конкретных технологических систем (СО – система освещения, СЭ – система электропитания, ОВ – система вентиляции и т.д.);

1. *Area*-названия конкретных систем/шкафов управления (ШРНН, ШУ ИТП и т.д.);

а. Название аварийного сигнала.

Пример: АРМ Э\ВНЕШНЯЯ ТЕРРИТОРИЯ\БКТП\СЭ\ШРНН\_1\IQF1.

### Извлечение данных из базы данных ALARMWORX64 LOGGER

В AlarmWorX64 Logger есть два основных типа аварийных сигналов – дискретные (Digital) и аналоговые (Limit). Сигналы могут быть в аварийном или нормальном состоянии. Аварийное состояние означает, что значение сигнала стало равно некоторой заданной величине, трактуемой как негативное, или сработало иное условие фиксации аварийного сигнала.

Дискретные сигналы имеют два состояния – норма и авария (рис. 1). Аналоговые имеют пять состояний (рис. 2): норма, *LoLo* (аварийно-низкий уровень), *Lo* (предупредительный низкий уровень), *Hi* (предупредительный высокий уровень),

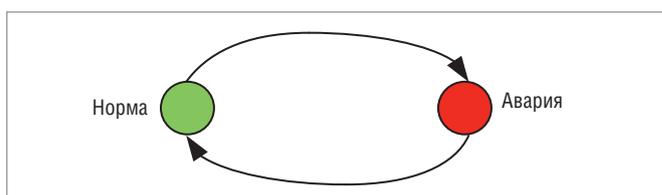


Рис. 1. Диаграмма изменения состояний дискретного аварийного сигнала

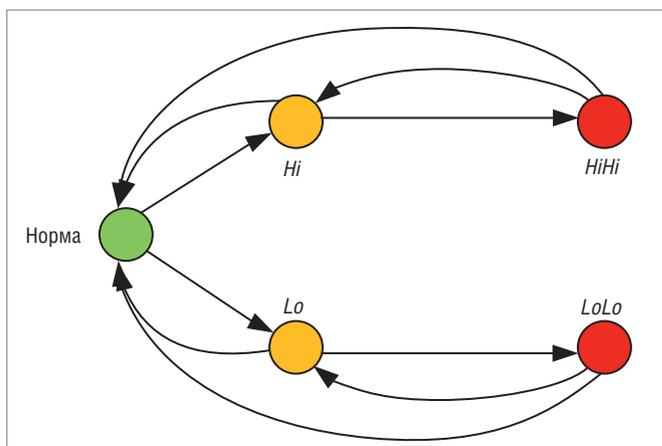


Рис. 2. Диаграмма изменения состояний аналогового аварийного сигнала

SubconditionName	Quality	AckRequired	Enabled	ServerProgID	EventTime	EventTimeMask	Source	ConditionName	Acked	ConditionActive	Message
126 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:01:40.000	021	Gamma 1	Digital	1	0	Gamma 1 Off
136 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:03:59.000	101	Alarm Test word 1	Limit	0	1	Test alarm word 1 Lo
137 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:05:45.000	085	Alarm Test word 1	Limit	0	1	Test alarm word 1 Hi
138 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:05:45.000	085	Alarm Test word 2	Limit	0	1	Test alarm word 2 Hi
139 IH	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:05:45.000	085	Alarm Test word 3	Limit	0	1	HiHi
140 enClient.DLL	192	0	0	ICONICS.AlarmSvr.1	2016-06-28 10:05:52.000	084	GenClient.DLL	The OPC Event Server is unavailable or does not...	0	0	The OPC Event Server is unavailable or does not...
141 enClient.DLL	192	0	0	ICONICS.AlarmSvr.1	2016-06-28 10:05:52.000	004	GenClient.DLL	Alarm subscription created successfully (ICONICS...	0	0	Alarm subscription created successfully (ICONICS...
142 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:05:52.000	795	Alarm Test word 1	Limit	0	1	Test alarm word 1 Hi
143 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:05:58.000	791	Alarm Test word 2	Limit	0	1	Test alarm word 2 Lo
144 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:06:01.000	193	Alarm Test word 3	Limit	0	1	Lo
145 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:06:05.000	097	TGF1	Digital	0	1	TGF1 On
146 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:06:07.000	196	Gamma 1	Digital	0	1	Gamma 1 On
147 enClient.DLL	192	0	0	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	086	GenClient.DLL	The OPC Event Server is unavailable or does not...	0	0	The OPC Event Server is unavailable or does not...
148 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	017	TGF1	Digital	0	1	TGF1 On
149 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	017	Alarm Test word 1	Limit	0	1	Test alarm word 1 Hi
150 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	017	Alarm Test word 2	Limit	0	1	Test alarm word 2 Lo
151 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	017	Alarm Test word 3	Limit	0	1	Lo
152 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:10.000	017	Gamma 1	Digital	0	1	Gamma 1 On
153 enClient.DLL	192	0	0	ICONICS.AlarmSvr.1	2016-06-28 10:07:11.000	140	GenClient.DLL	Alarm subscription created successfully (ICONICS...	0	0	Alarm subscription created successfully (ICONICS...
154 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:19.000	023	Alarm Test word 1	Limit	0	1	Test alarm word 1 Lo
155 D	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:22.000	025	Alarm Test word 2	Limit	0	0	Test alarm word 2 Normal
156 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:07:24.000	126	Alarm Test word 3	Limit	0	1	Hi
157 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:14:12.000	020	TGF1	Digital	0	1	TGF1 On
158 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:16:57.000	114	TGF1	Digital	0	1	TGF1 On
159 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:17:05.000	016	TGF1	Digital	0	0	TGF1 Off
160 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:17:13.000	024	TGF1	Digital	0	1	TGF1 On
161 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:17:34.000	343	TGF1	Digital	0	1	TGF1 On
162 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:18:12.000	779	TGF1	Digital	0	1	TGF1 On
163 enClient.DLL	192	0	0	ICONICS.AlarmSvr.1	2016-06-28 10:18:52.000	052	GenClient.DLL	The OPC Event Server is unavailable or does not...	0	0	The OPC Event Server is unavailable or does not...
164 igit	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:18:57.000	046	Gamma 1	Digital	0	1	Gamma 1 On
165 I	192	1	1	ICONICS.AlarmSvr.1	2016-06-28 10:18:57.000	017	Alarm Test word 1	Limit	0	1	Test alarm word 1 Hi

Рис. 3. Фрагмент базы данных AlarmWorX64 Logger

*HiHi* (аварийно-высокий уровень). Точная трактовка аварийного состояния зависит от уровня *Severity* (уровня критичности сигнала), задаваемого для каждого аварийного состояния, например: 100 – рядовое событие; 500 – предупреждение о предаварийном состоянии типа загрязнённый фильтр; 700 – критичная авария типа потери питания/неисправности оборудования. Иногда для конкретной установки уровень *Lo* – это не предупредительный низкий, а аварийный сигнал о снижении какого-либо параметра, и его уровень значимости будет высоким. Уровень *Severity* удобно использовать для фильтрации сигналов, чтобы избежать попадания в отчётную документацию малозначащих сигналов. В разработанной системе требовалось заносить в отчёт все сигналы с уровнем *Severity* выше 500.

Каждый переход от нормального состояния к аварии (на рис. 1, 2 показаны стрелками) фиксируется в БД AlarmWorX64 Logger. В случае перехода от нормального состояния к аварии требуется квитирование такого перехода – подтверждение оператором фиксации данной информации. Факт квитирования заносится в БД AlarmWorX64 Logger и может использоваться для отчёта.

На рис. 3 приведён фрагмент данных из БД AlarmWorX64 Logger. В БД попадают *Area*, название аварийного сигнала, время изменения состояния, комментарий при квитировании, время квитирования, текст описания аварийного сигнала, признак активности/квитирования аварийного сигнала, битовая маска изменения состояния аварийного сигнала и множество другой, по большей части служебной информации.

Для извлечения данных из БД AlarmWorX64 Logger необходимо написать SQL-запросы. В ReportWorX имеется редактор запросов к БД и в большинстве случаев его вполне достаточно для построения простых запросов. Но при необходимости построения сложных запросов, как в случае поставленной задачи, с вложенными подзапросами и предварительной обработкой извлечённых данных, встроенного редактора уже недостаточно. Поэтому было принято решение создать

запрос внутри БД AlarmWorX64 Logger – специализированную хранимую процедуру *spGetTodayAlarms*, которая автоматизирует выборку и подготовку данных для отчётов. Текст хранимой процедуры приведён в листинге 1.

Процедура *spGetTodayAlarms* получает на вход следующие параметры: *@SystemName* – название системы, *@ARMName* – название АРМ, на который выводится информация по системе, *@SeverityVal* – уровень значимости сигнала. На выходе процедуры таблица, содержащая следующие колонки: название объекта/сооружения, название системы, название аварийного сигнала, текст аварийного сообщения, время возникновения аварии, время квитирования, время снятия аварии.

Внутри процедуры *spGetTodayAlarms* автоматически определяется временной диапазон от начала до конца текущего дня, и на его основе формируется временная таблица *#TempAlarmLog*, содержащая данные из БД AlarmWorX64 Logger только за текущий день. А уже из временной таблицы выполняется выборка, увязывающая цепочку событий: появление аварийного сигнала, квитирование и снятие аварийного сигнала.

Здесь следует отметить важность колонки *ChangeMask* в БД AlarmWorX64 Logger, которая содержит битовую маску, однозначно определяющую, какие изменения произошли с аварийным сигналом (в том числе квитирование). На основании сочетания (листинг 1) значения этой колонки, колонки *ConditionActive* (сработало условие фиксации аварийного сигнала) и *Acked* (квитирование аварийного сигнала) определяется момент возникновения аварийного сигнала, его квитирования и снятия аварийного сигнала. Более подробную информацию по этим колонкам можно найти в документации на GENESIS64.

Название объекта/сооружения и название конкретной системы формируются путём разбора данных колонки *Area* и наполнения указанных колонок только нужной информацией. Это необходимо, поскольку из-за сложной иерархии *Area* в AlarmWorX64 Server в БД AlarmWorX64 Logger записываются

очень длинные пути в поле *Area* и без их разбора колонки отчёта будут слишком длинными и будут содержать ненужную информацию. Например:

Исходное значение *Area* – АРМ Э\ВНЕШНЯЯ ТЕРРИТОРИЯ\БКТП\СЭ\ШРНН\_1.

После разбора колонка с названием объекта будет содержать «ВНЕШНЯЯ ТЕРРИТОРИЯ\БКТП», а колонка с названием системы будет содержать «ШРНН\_1».

Также в *spGetTodayAlarms* происходит анализ собранной информации, и в соответствующих полях (с метками времени) выводится поясняющая информация. Например, если ава-

рийный сигнал не квитирован на момент формирования отчёта, в поле «время квитирования» будет написано «Не квитировано». Если аварийный сигнал ещё не пропал или для аналогового сигнала произошло изменение одного аварийного уровня на другой, будет написано «Авария не снята/сменила уровень».

## СОЗДАНИЕ ОТЧЁТА ДЛЯ REPORTWORX

Для создания отчёта необходимо в ReportWorX Configurator создать новую конфигурацию (рис. 4) или использовать имеющуюся. Для конфигурации важно задать имя, и путь к

### Листинг 1. Функция spGetTodayAlarms

```
USE [AlarmWorX64 Logger Database]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[spGetTodayAlarms]
-- Add the parameters for the stored procedure here
@SystemName varchar(100) = '',
@ARMName varchar(100) = '',
@SeverityVal int = 500
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets
from
-- interfering with SELECT statements.
SET NOCOUNT ON;
SET LANGUAGE russian
DECLARE @dd1 datetime, @dd2 datetime

-- Insert statements for procedure here
SET @dd1 = DATEADD(hh,00,GETDATE()- {fn
CURRENT_time()}) -- Начало текущего дня
SET @dd2 = cast(convert(char(10),@dd1,112)+'23:59:59'
as datetime) -- Конец текущего дня

SET @ARMName = @ARMName + '\'
SET @SystemName = '\' + @SystemName
/*Создаём временную таблицу для выборки алармов за
сегодня*/
SELECT [Area],[EventTime],[Source],[Acked],[Condi-
tionActive],[Message]
,[ActiveTime],[ChangeMask]
INTO #TempAlarmLog
FROM [AlarmWorX64 Logger Database].[dbo].[AlarmLog]
WHERE [Area] LIKE @ARMName + '%' + @SystemName + '%'
and Severity >= @SeverityVal
and [_AckRequired] = 1 AND ([EventTime] BETWEEN
@dd1 AND @dd2)

/* Делаем выборку из временной таблицы, увязывая це-
почку событий:
появление аларма, квитирование, пропажа аларма */
SELECT SUBSTRING(
SUBSTRING(
EventTimeTable.Area,
CHARINDEX( '\', EventTimeTable.Area) + 1,
LEN(EventTimeTable.Area)), 1,
```

```
CHARINDEX(
@SystemName,
SUBSTRING(
EventTimeTable.Area,
CHARINDEX( '\', EventTimeTable.Area) + 1,
LEN(EventTimeTable.Area))) - 1
) As ObjectName,
SUBSTRING(EventTimeTable.Area, CHARINDEX(@System-
Name, EventTimeTable.Area) +
LEN(@SystemName) + 1,
LEN(EventTimeTable.Area)) As SystemName,
EventTimeTable.Source,
EventTimeTable.Message,
REPLACE(ISNULL(convert(char(100),
EventTimeTable.EventTime, 113),
'Не известно'),':000', '') As EventTime,
REPLACE(ISNULL(convert(char(100),
AckedTimeTable.EventTime, 113),
'Не квитирована'),':000', '') As AckedTime,
REPLACE(ISNULL(convert(char(100), Condition-
TimeTable.EventTime, 113),
'Авария не снята/сменила уровень'),':000', '')
As ConditionTime
FROM
(SELECT *
FROM #TempAlarmLog
WHERE [Acked] = 0 And [ConditionActive] = 1) As
EventTimeTable
LEFT JOIN (SELECT
[EventTime],[ActiveTime],[Acked],[ChangeMask],[Condi-
tionActive]
FROM #TempAlarmLog
WHERE ([Acked] = 1 And [ConditionActive]= 1 And
[ChangeMask] = 2) OR
([Acked] = 1 And [ConditionActive]= 0 And
[ChangeMask] = 2)) As AckedTimeTable
ON (EventTimeTable.ActiveTime = AckedTimeTable.Ac-
tiveTime)
LEFT JOIN (SELECT [EventTime],[ActiveTime],[Change-
Mask],[ConditionActive]
FROM #TempAlarmLog
WHERE ([ConditionActive]= 0) And
([ChangeMask] = 1 OR [ChangeMask] = 257)) As Con-
ditionTimeTable
ON (EventTimeTable.ActiveTime = Condition-
TimeTable.ActiveTime)

DROP TABLE #TempAlarmLog;
END
```



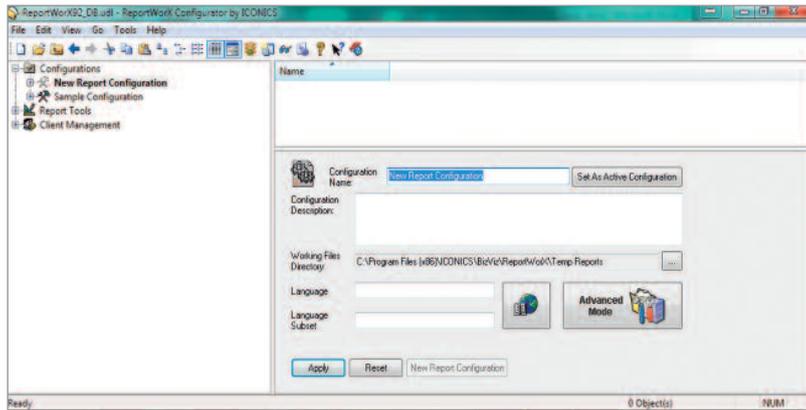


Рис. 4. Создание новой конфигурации в ReportWorX Configurator

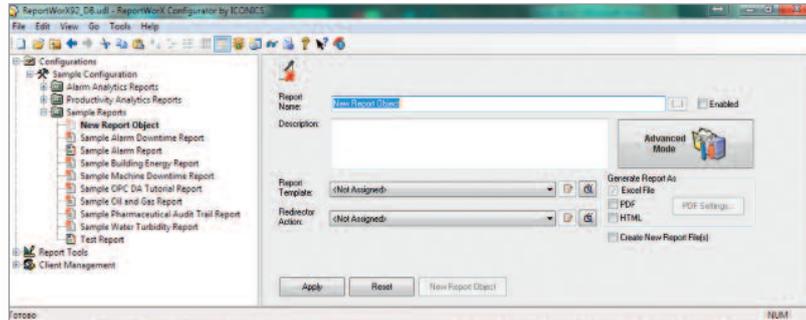


Рис. 5. Создание нового отчёта в ReportWorX Configurator

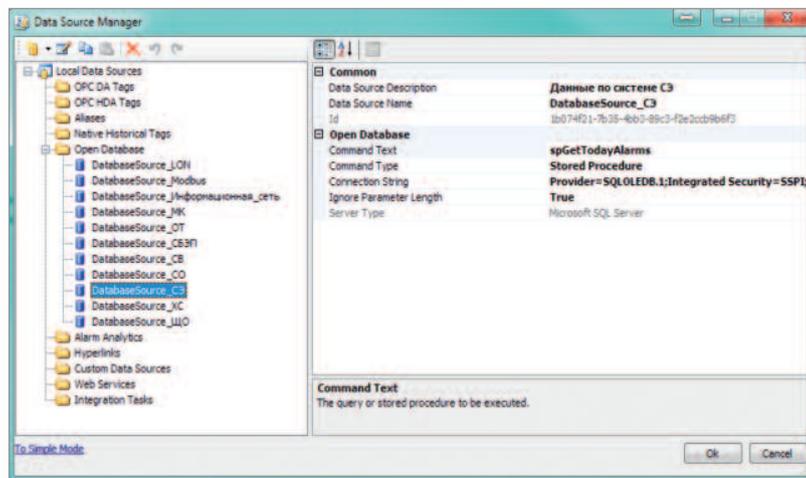


Рис. 6. Окно настройки источников данных для шаблона ReportWorX

папке для временных файлов отчётов. Далее создаётся сам отчёт (Report Object). Для него можно задать (рис. 5) имя, шаблон и действия по распространению отчёта: пересылка по почте, публикация в Интернете, сохранение копии отчёта на файловом ресурсе. После создания Report Object можно перейти к созданию шаблона отчёта (Report Template), для этого необходимо нажать на кнопку “Create new template object” напротив надписи “Report Template” (рис. 5). Запущенный мастер предложит задать имя шаблона, папку для размещения шаблона, формат файла шаблона (Excel 2003/Excel 2007), а также предложит скопировать форматирование, ссылки на источники данных из других шаблонов. После этого будет запущена программа Excel с надстройкой ReportWorX.

На открывшемся листе Excel оформляется внешний вид отчёта, указывается заголовок, поля для автоматической вставки времени и даты и т.д. После этого выполняется привязка к шаблону источников данных, для чего на вкладке ReportWorX

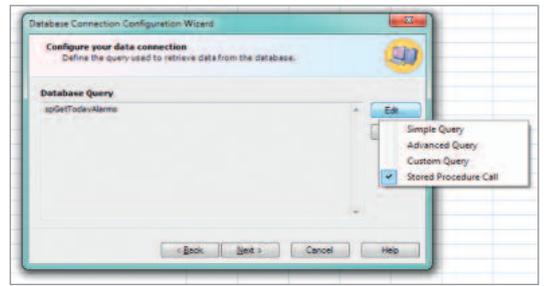


Рис. 7. Выбор типа запроса к базе данных

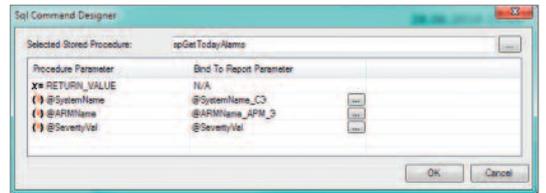


Рис. 8. Окно настройки параметров хранимой процедуры

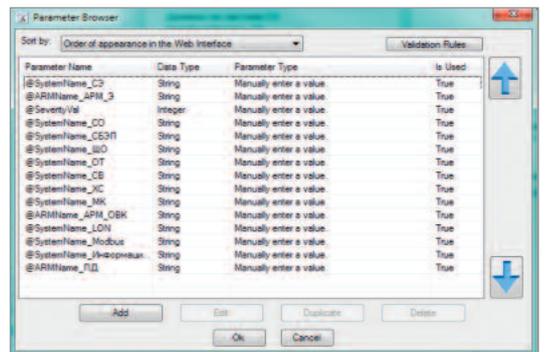


Рис. 9. Перечень переменных отчёта

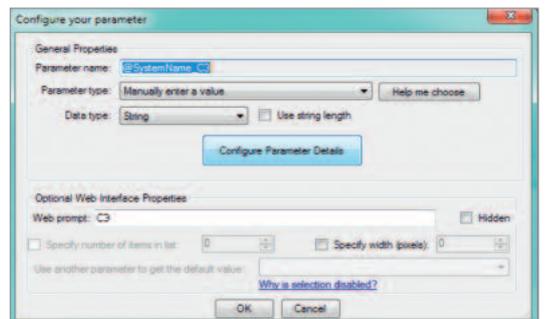


Рис. 10. Параметры переменной отчёта

на ленте с инструментами необходимо нажать кнопку “Data Source Management”, откроется окно для настройки источников данных (рис. 6): это могут быть теги OPC DA, OPC HDA, базы данных (БД), гиперссылки и т.д. В данном проекте источником данных выступала база данных AlarmWorX64 Logger.

Для подключения к БД AlarmWorX64 Logger необходимо выбрать в списке “Local Data Source” пункт “Open Database” и через контекстное меню запустить мастер создания соединения с БД. Первый шаг мастера – подключение к серверу БД и выбор базы данных на сервере.

Следующий шаг – формирование запроса к БД (ссылки на данные), который извлечёт нужные для отчёта данные. Как было описано ранее, для извлечения данных из БД AlarmWorX64 Logger будет использоваться хранимая процедура spGetTodayAlarms, поэтому при нажатии на кнопку “Edit” выбираем из списка пункт “Stored Procedure Call” (рис. 7). В открывшемся окне (рис. 8) при нажатии на кнопку “...” (справа сверху) выбирается нужная хранимая процедура.

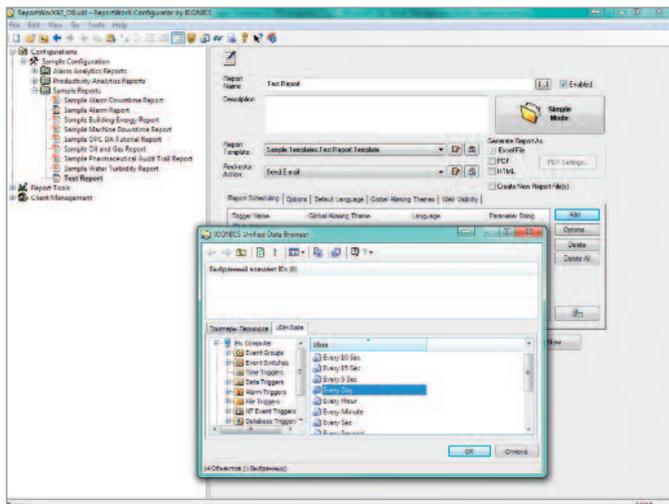


Рис. 11. Настройка периодичности генерации отчёта

После этого в таблице под названием процедуры отображается перечень параметров хранимой процедуры. Эти параметры необходимо сопоставить так называемым переменным отчёта. Для этого надо нажать кнопку “...” напротив каждой переменной. В открывшемся окне (рис. 9) отображается список переменных отчёта, при нажатии кнопки Add добавляется новая переменная. Для каждой переменной (рис. 10) указывается имя, тип данных и значение по умолчанию. После настройки переменной она выбирается в списке переменных отчёта и при нажатии кнопки “Ок” (рис. 9) сопоставляется с параметром хранимой процедуры. Теперь при генерации отчёта хранимая процедура будет получать на вход данные

из указанных переменных. Их значение может задаваться в ReportWorX Configurator или в Web-интерфейсе ReportWorX. Если значения не заданы, то будут использованы значения по умолчанию.

Последнее оказалось очень удобно, поскольку необходимо на разных вкладках отчёта выводить информацию по разным системам, а хранимая процедура одна, и возник вопрос, как передать параметры, однозначно определяющие технологическую систему, в хранимую процедуру. Для каждой технологической системы были созданы индивидуальные запросы на базе spGetTodayAlarms и созданы соответствующие переменные отчёта, а им назначены значения по умолчанию, однозначно определяющие нужную систему. Тогда при генерации отчёта каждый раз формировались отдельные запросы с нужными параметрами, но при этом в БД AlarmWorX64 Logger использовалась только одна хранимая процедура.

После завершения настройки ссылок на данные в шаблоне выделяется область ячеек для вывода данных, полученных по запросу. Размер области зависит от количества полей, возвращаемых запросом. Используя условное форматирование (доступно в Excel 2007 и выше), зависящее от содержимого ячейки, можно сделать более наглядное оформление отчёта, например, подсветив строки с авариями, которые не квитированы или активны.

После завершения оформления шаблона отчёта при закрытии Excel возникнет диалоговое окно с предложением сохранить сделанные изменения. Сохранив изменения, можно перейти к следующему этапу — настройке периода генерации отчёта. Для этого необходимо выбрать отчёт и в области его настроек (рис. 5) нажать “Advanced Mode”, после этого выбрать

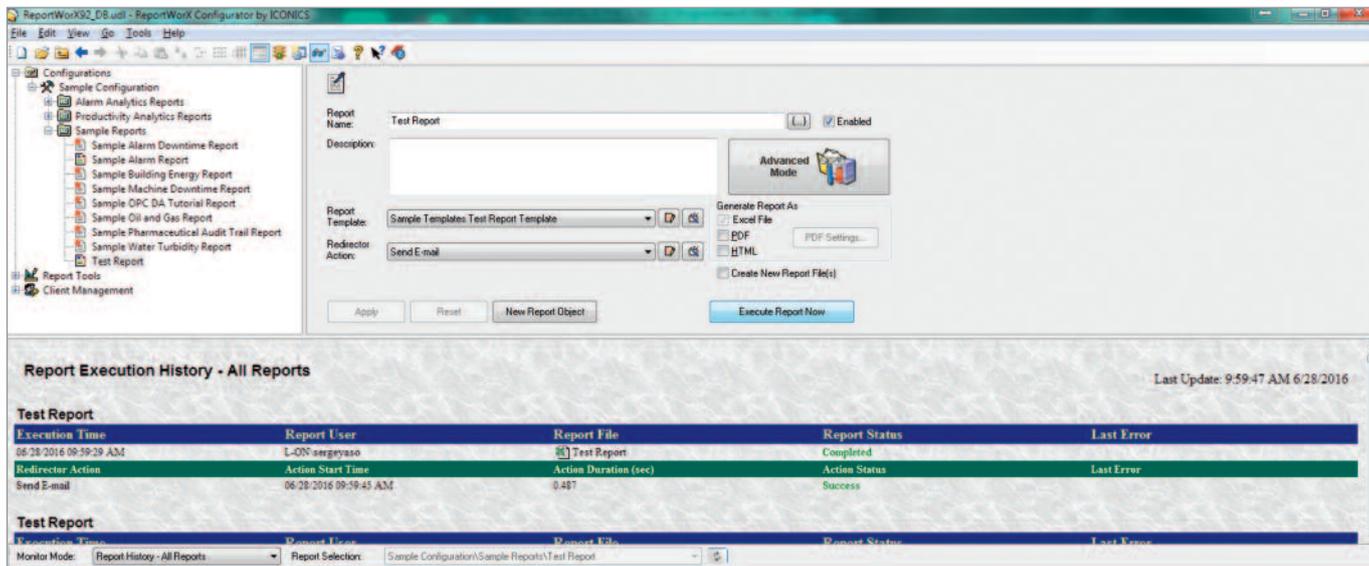


Рис. 12. Проверка генерации отчёта

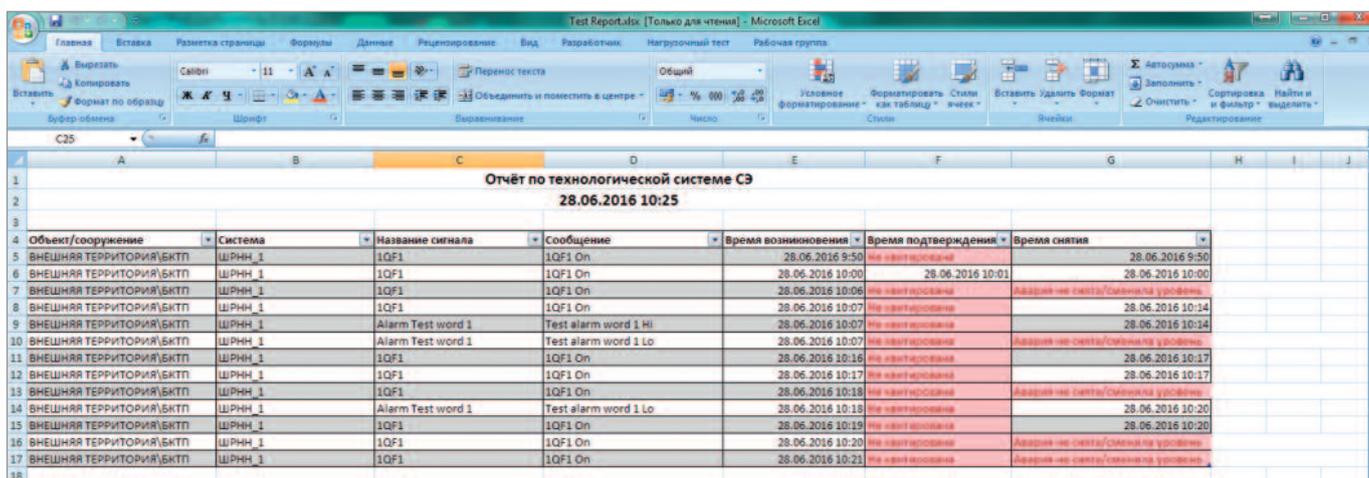


Рис. 13. Сгенерированный отчёт

вкладку Report Scheduling (рис. 11) и нажать кнопку “Add...”, выбрать Time Triggers и триггер с нужным периодом генерации отчётов. Поскольку нужного триггера в 24 часа по умолчанию нет, то он был создан вручную в приложении UDMCf (Unified Data Manager).

После создания отчёта можно проверить его генерацию, нажав кнопку “Execute Report Now” (рис. 12). На рис. 13 показан пример сгенерированного отчёта. В нём интересно отметить не характерное для дискретного аварийного сигнала повторение описания «Авария не снята/сменила уровень», поскольку он не может сменить уровень без перехода в нормальное состояние, которого в отчёте нет. Это связано с перезагрузкой сервера AlarmWorX64 Server, после которой все аварии обрабатываются как вновь появившиеся, что приводит к дублированию в отчёте аварийных сигналов.

Бороться с этим нежелательно, так как достоверно неизвестно, «новая» это авария или повторно зарегистрированная старая.

Если предварительно в ReportWorX Configurator открыть Monitor View (кнопка на панели в виде очков), то внизу откроется протокол работы ReportWorX, в котором можно увидеть статус генерации отчёта. Там же можно открыть файл отчёта, щёлкнув по значку файла левой кнопкой мыши.

После настройки отчёта не стоит забывать включить его – флажок “Enabled” на рис. 12.

### НАСТРОЙКА РАССЫЛКИ ОТЧЁТОВ

Отчёты используются не только оперативным персоналом, но и руководством, поэтому важно иметь возможность рассылки отчётов заинтересованным лицам. Для этого в ReportWorX предусмотрены так называемые Redirector Actions, действия для распространения отчётов. В описываемой задаче интересовал только один тип Redirector Actions – E-mail Actions, рассылка по электронной почте. Для настройки данного действия нужно выбрать отчёт и в области его настроек нажать кнопку “Create new RedirectorTask”, будет запущен мастер настройки пересылки отчёта. При выборе E-mail Report будет создано действие (Action) для пересылки по электронной почте, далее необходимо заполнить традиционные для любого электронного письма поля (рис. 14): адресат, тема письма, текст письма. Для работы рассылки по электронной почте также надо задать SMTP-настройки системы (рис. 15).

Следует обратить внимание, что для отправки по умолчанию используется 25-й порт SMTP-сервера, если необходимо изменить порт для отправки, необходимо внести в файл C:\ProgramData\ICONICS\IcoSetup.ini, строки:

```
[ReportWorX.NET\ReportWorXRedirector]
SMTPPort=2525
```

Если требуется включить SSL при отправке почты, то в файл настроек в секцию [ReportWorX.NET\ReportWorXRedirector] надо внести строку:



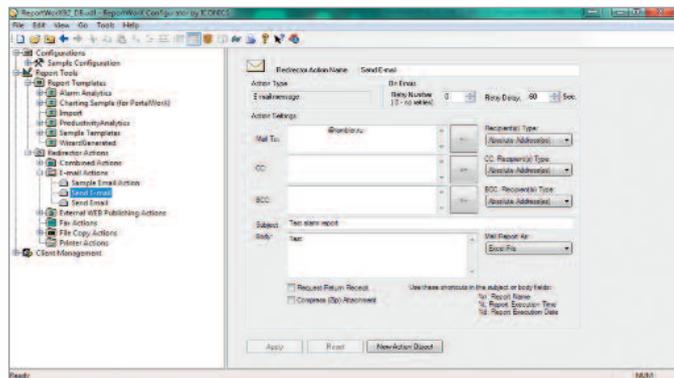


Рис. 14. Настройка рассылки отчёта по электронной почте

*EnableSSLForEmail=1*

После изменения настроек необходимо перезапустить ReportWorX-сервис – кнопка в виде светофора в ReportWorX Configurator. Теперь в конце каждого дня при срабатывании триггера генерации отчёта будет происходить автоматическая рассылка.

### Возможности расширения созданной системы

За счёт использования современных программных продуктов ICONICS созданная система периодических отчётов получилась расширяемой и масштабируемой. И компания «Инсайт-НГ» уже готова предложить заказчику возможное направление развития данной системы.

В июне 2015 года компания ICONICS представила новую версию надстройки ReportWorX Express, которая даёт возможность

пользователям платформы Microsoft Office получать доступ к данным из приложений ICONICS (GENESIS64, Hyper Historian и AnalytX и др.).

Обновлённый продукт ICONICS ReportWorX Express может работать как в классическом приложении Microsoft Excel 2013, так и в веб-приложении Excel Online для облачной службы Microsoft Office 365.

С помощью ReportWorX Express можно быстро и просто настраивать и просматривать отчёты по производственным процессам. Встроенная система безопасности реализует управление доступом пользователей к данным на основе ролей. Интуитивно понятный интерфейс позволяет легко выбирать необходимые данные и определять формат их отображения на листе Excel, а затем применять нужные фильтры.

Такое решение в перспективе может оказаться очень интересным, особенно для заказчиков с территориально распределёнными системами, например, общероссийского масштаба, поскольку позволяет снизить издержки на развёртывание системы и расходы на лицензирование. Также оно может быть полезным и для небольших заказчиков, так как за счёт поддержки Office 365 не требует установки программ из пакета Microsoft Office на рабочих местах. ●

**E-mail: ssa-company@rambler.ru**

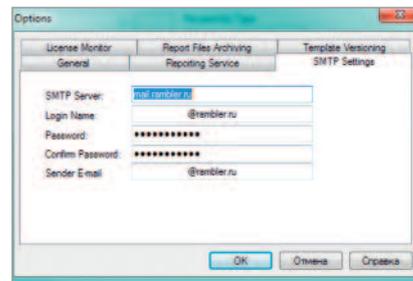


Рис. 15. SMTP-настройки ReportWorX

## НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ

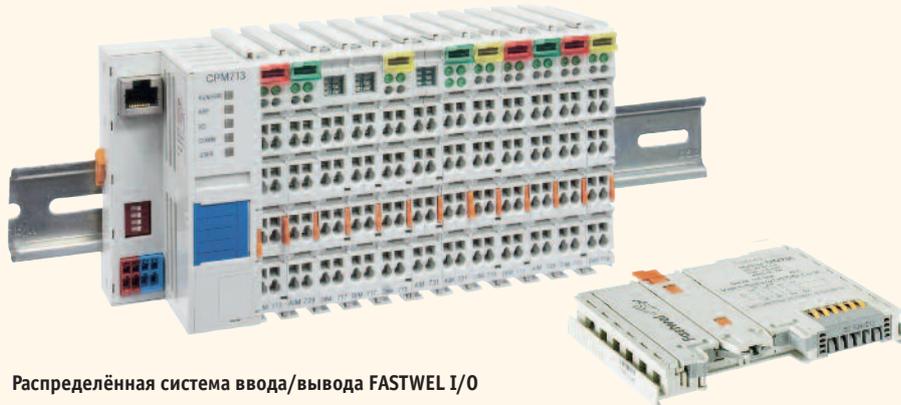
### Контроллер FASTWEL I/O готов для использования в системах пожарной автоматики

Распределённая система ввода/вывода FASTWEL I/O получила сертификат соответствия в области пожарной безопасности. Согласно данному сертификату, система соответствует требованиям ГОСТ Р 53325-2012 «Техника пожарная. Технические средства пожарной автоматики. Общие технические требования и методы испытаний».

Благодаря полученному сертификату контроллеры FASTWEL I/O теперь могут использовать системные интеграторы, разрабатывающие и проектирующие АСУ ТП в области пожарной сигнализации, автоматического пожаротушения, жизнеобеспечения зданий и промышленных объектов.

Дополнительным преимуществом для использования системы FASTWEL I/O в системах пожарной автоматики является наличие модулей дискретного и аналогового ввода с контролем целостности цепей. Такие модули подходят для подключения неадресных извещателей пожарной и охранной сигнализации.

В итоге при использовании FASTWEL I/O можно построить систему пожарной автоматизации, которая обеспечивает:



Распределённая система ввода/вывода FASTWEL I/O

- интеграцию системы в сеть предприятия и передачу данных по протоколам CANopen, MODBUS RTU/TCP, DNP3 и PROFIBUS DP-V1;
- приём сигналов от извещателей пожарной и охранной сигнализации;
- контроль состояния линии связи и узлов;
- индикацию, отображающую текущее состояние каждого канала;
- ведение журнала событий.

FASTWEL I/O – российская распределённая система ввода-вывода, созданная с учётом требований отечественных потреби-

лей. Идеология FASTWEL I/O основана на предоставлении разработчику максимальных возможностей в конфигурировании, наращивании и обслуживании системы.

Система FASTWEL I/O всего за несколько лет заслужила доверие заказчиков в самых разных отраслях. Растущая популярность контроллеров наглядно демонстрирует российским системным интеграторам, что отечественный продукт по ряду параметров, таким как функциональность, надёжность и доступность, превосходит зарубежные аналоги. ●



## Компания ПРОСОФТ признана лучшим дистрибьютором MEN



Компания ПРОСОФТ получила звание лучшего дистрибьютора продукции MEN

Компания MEN Mikro Elektronik провела ежегодную встречу партнёров Partner Connections Summit 2016, в рамках которой компания ПРОСОФТ – эксклюзивный дистрибьютор продукции MEN на территории России и стран СНГ – получила почётное звание «Лучший дистрибьютор».

На ежегодном мероприятии MEN Partner Connections Summit, состоявшемся в Нюрнберге, где расположена штаб-квартира известного производителя электроники, для подведения итогов работы и представления новинок собрались более 10 ключевых дистрибьюторов из стран Европы и Азии.

В рамках встречи компания ПРОСОФТ получила почётное звание «Лучший дистрибьютор» – награду вручал Бернд Хэртглийн, главный операционный директор MEN, а получил её бренд-менеджер Алексей Пятничкиных.

Компания ПРОСОФТ начала сотрудничать с MEN в 2010 году, выбрав её в качестве надёжного поставщика оборудования для железнодорожного транспорта, авиации и промышленной автоматизации. Тот факт, что вся продукция MEN предназначена для работы в самых жёстких условиях, обусловил постоянно растущий спрос на неё в России.

Большую популярность решения MEN завоевали в Северо-Западном регионе РФ, в том числе благодаря слаженным действиям подразделения ПРОСОФТ-Санкт-Петербург, которое провело большую работу по продвижению продукции MEN. Свою роль сыграл также выпуск новинок MEN, выполненных по новейшему стандарту CompactPCI Serial, которые оказались чрезвычайно востребованными в регионе.

Начиная с 2010 года компания ПРОСОФТ в рамках продвижения бренда MEN провела большое количество технических семинаров по продукции этой компании в разных городах РФ, а также приняла участие в авторитет-

ных специализированных выставках. Так, в 2015 году продукция MEN была представлена на крупнейших отраслевых мероприятиях, таких как Международный железнодорожный салон EXPO 1520, Международный авиационно-космический салон МАКС-2015 и Международный военно-морской салон.

Отметим, что в начале октября 2016 года в Санкт-Петербурге состоится очередной семинар, посвящённый продукции MEN, на котором ПРОСОФТ продолжит знакомить отечественных разработчиков с передовыми техническими решениями из мира встраиваемых систем. ●

## CompactPCI на Марсе

В этом году марсоход Opportunity отмечает своё 12-летие на Красной планете. Посадка марсохода состоялась 24 января 2004 года, и он до сих пор продолжает функционировать.

Opportunity совершил посадку в кратере Игл на плато Меридиана. В настоящее время он находится в районе кратера Индевор,

пройдя более 40 км от своего первоначального положения.

Марсоход управляется двумя компьютерами на базе стандарта CompactPCI, спроектированными и построенными инженерами компании BAE Systems.

Когда Opportunity приземлился, команда NASA думала, что суровый марсианский климат сделает его неработоспособным в течение нескольких месяцев. Запланированный срок работы марсохода составлял примерно 90 суток. Но космический ветеран, получающий энергию только от солнечных батарей, всё ещё собирает данные.

Атмосфера Марса более разрежена, чем воздушная оболочка Земли, и на 95,9% состоит из углекислого газа, около 1,9% приходится на долю азота, и 2% составляет аргон, содержание кислорода всего 0,14%. Среднее давление атмосферы на поверхности планеты в 160 раз меньше, чем у поверхности Земли.

Средняя температура на Марсе значительно ниже, чем на Земле, – около  $-40^{\circ}\text{C}$ . При наиболее благоприятных условиях летом на дневной стороне планеты воздух прогревается до  $+20^{\circ}\text{C}$  – вполне приемлемая температура для жителей Земли. Но зимней ночью мороз может достигать  $-125^{\circ}\text{C}$ , при такой температуре даже углекислота замерзает, превращаясь в сухой лёд.

CompactPCI является жизнеспособной и важной технологией для широкого спектра применений. Вычислительные модули, выполненные согласно стандарту CompactPCI, используются для построения спутников и для других космических применений, таким образом доказывая свою надёжность и работоспособность. ●



Фото NASA