

УТВЕРЖДЕН

SIMBIX.00003-01 13 ЛУ

ООО «ЭЛЕКТРОННОЕ ПРОЕКТИРОВАНИЕ»

(наименование организации разработчика)

SIMBIX.ИНТЕГРАЦИОННЫЙ ПОРТАЛ

(наименование программы)

ВЕРСИЯ 1.0

(версия программы)

SIMBIX.IP

(условное обозначение программы)

ОПИСАНИЕ ПРОГРАММЫ

(наименование документа)

SIMBIX.00003-01 13

(условное обозначение документа)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В данном документе приводятся общие сведения о программе Simbix. Интеграционный портал, функциональное назначение программы, ее логическая структура, необходимые программные и технические средства для функционирования программы, вызов и загрузка программы, а также входные и выходные данные.

Перв. применен.

Справ. №

Подпись и дата

Инв. № дубл.

Взам. инв. №

Подпись и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Ясиновская Е.Д.		
Провер.				
Т. контр.				
Н. контр.				
Утв.				

Simbix. Интеграционный портал

Описание программы

Лит.	Лист	Листов
	2	25

ООО «Электронное
Проектирование»

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ.....	4
1.1	Обозначение и наименование программы	4
1.2	Программное обеспечение	4
1.3	Языки программирования, на которых написана программа	4
2	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	5
3	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	7
3.1	Алгоритм программы.....	7
3.2	Используемые методы	8
3.3	Структура программы.....	10
3.4	Описание функций программы.....	12
3.4.1	Функция «s2CopyNode»	12
3.4.2	Функция «s2CountNodes»	12
3.4.3	Функция «s2DeleteNode»	12
3.4.4	Функция «s2Fetch»	12
3.4.5	Функция «s2GetBranch».....	13
3.4.6	Функция «s2GetChildren»	13
3.4.7	Функция «s2GetCurrentUser»	14
3.4.8	Функция «s2GetKnee»	14
3.4.9	Функция «s2GetNode»	14
3.4.10	Функция «s2Chmod».....	15
3.4.11	Функция «s2Chown».....	15
3.4.12	Функция «s2GetNodeProto»	15
3.4.13	Функция «s2GetNodes»	15
3.4.14	Функция «s2GetParent»	16
3.4.15	Функция «s2GetPath».....	16
3.4.16	Функция «s2Hash».....	17
3.4.17	Функция «s2Hex2Str»	17
3.4.18	Функция «s2ImageResize».....	17
3.4.19	Функция «s2IndexArray».....	17
3.4.20	Функция «s2InsertNode»	17
3.4.21	Функция «s2Login».....	18
3.4.22	Функция «s2Logout».....	18
3.4.23	Функция «s2MoveNode».....	18
3.4.24	Функция «s2Php2Js»	18
3.4.25	Функция «s2SetOrder»	18
3.4.26	Функция «s2SortNodes»	19
3.4.27	Функция «s2Str2Hex»	19
3.4.28	Функция «s2UpdateNode»	19
3.4.29	Функция «append»	19
3.4.30	Функция «explode»	20
3.5	Связи программы с другими программами	20
4	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	21
5	ВЫЗОВ И ЗАГРУЗКА	22
6	ВХОДНЫЕ ДАННЫЕ.....	23
7	ВЫХОДНЫЕ ДАННЫЕ.....	24

Име. № подл.			
Подпись и дата			
Взам. име. №			
Име. №			
Подпись и дата			

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Simbix.Интеграционный портал – это программная платформа для сопряжения многопользовательских информационных систем (ИС) и обеспечения полноценного информационного обмена и взаимодействия между ними на основе единой модели данных (ЕМД).

Simbix.Интеграционный портал используется для построения единого информационного пространства, в котором на основе семантического, нормативно-правового и организационно-методического единства обеспечивается взаимодействие между информационными системами различной направленности. К таким системам могут относиться корпоративные (финансовые, учетные, аналитические, ERP, CRM и др.) или ведомственные информационные системы федеральных/региональных органов власти.

Внедрение платформы «Интеграционный портал» позволяет обеспечить не только публикацию данных из различных ИС в единой точке доступа к информации, но и предоставляет возможность реализовать различные шаблоны межсистемных поисковых запросов, благодаря применению ЕМД.

Simbix.Интеграционный портал обеспечивает:

- инструментарий сбора и систематизации первичных сведений и исходных данных для формирования единого информационного пространства;
- взаимодействие с программной платформой EP.Экспертная система;
- экспорт исходных данных из Simbix.Интеграционный портал для проведения экспертного анализа в формате XML в асинхронном режиме.
- импорт результатов анализа в форматах XML, XSD, BPEL в асинхронном режиме для представления Simbix.Интеграционный портал;
- визуальное представление результатов, получаемых из EP.Экспертная система, для последующей верификации;
- представление единой модели данных для участников информационного взаимодействия;
- распространение метаданных для последующей интеграции информационных систем;
- коллективную работу широкого круга экспертов, участвующих в формировании информационного пространства, включая возможность комментирования и формирования экспертного сообщества;
- формирование библиотек информационных и методических материалов для формирования единого информационного пространства;

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						5

- средства публикации новостной, текстовой и графической информации, включая настройку визуальных интерфейсов;
- настройку публикации сведений из интегрированных информационных систем;
- настройку механизмов организации поиска в интегрированных информационных системах на основе единой модели данных;
- настройку учетных записей пользователей и возможность разграничения прав пользователей.

Инев. № подл.		Подпись и дата		Взам. инв. №		Инев. №		Подпись и дата	
Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13				Лист
									6

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Алгоритм программы

Программа Simbix построена на принципах объектно-ориентированного подхода и состоит из двух основных компонентов: программного интерфейса управления данными и системы управления корпоративным контентом. Основными объектами программы являются узлы и модули данных.

Первый компонент позволяет создавать на различных логических страницах проекта произвольные типы узлов данных и связи между ними, а также позволяет управлять отображением созданных узлов через веб-интерфейс. Описание узлов и модулей данных и связи между ними размещаются в объектном хранилище.

Второй компонент предназначен для управления узлами как информационными элементами.

Узлы данных объединяются (группируются) в модули данных и связываются между собой в зависимости от требований бизнес-логики конкретного проекта управления данными.

Управление информацией в созданном проекте осуществляется на уровне иерархии узлов, каждый из которых может быть расположен одновременно на нескольких логических страницах.

Структура информации, хранящейся в объектах, определяется набором классов, определенных разработчиком на этапе проектирования системы и может быть изменена без ущерба работающего проекта в любой момент.

Возможностью создания произвольных типов объектов, настройка связи между ними, а также управление отображением объектов в через веб-обозреватель позволяет просто, оперативно и гибко решать задачи, связанные с изменением структуры создаваемых данных.

В программе предусмотрена работа трех групп пользователей – администраторы, редакторы и клиенты. Доступ к программе и отнесение пользователей к различным группам осуществляются с применением процедур авторизации и аутентификации пользователя.

Для работы пользователей групп администратора и редактора используется административный интерфейс, который запускается с использованием веб-браузера на удаленном компьютере, пользователи группы клиентов используют исключительно интерфейс отображения данных через стандартный тонкий клиент (веб-браузер), не имея доступа к администраторским функциям.

Администраторы используют функционал для настройки (конфигурирования) программы, редакторы используют функционал управления бизнес-логикой программы и ввода информации.

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата
--------------	----------------	--------------	--------	----------------

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						7

В режиме администратора программа позволяет выполнять следующие действия по изменению метаданных и данных системы:

- создание нового модуля данных;
- создание нового типа в модуле данных;
- создание нового метода;
- изменение реквизитов модуля данных;
- изменение реквизитов типа данных;
- изменение содержания метода;
- удаление метода;
- удаление типа данных;
- удаление модуля данных;
- создание новой группы пользователей;
- создание новой учетной записи уровня редактора;
- привязка новой учетной записи к произвольной группе пользователей;
- изменение реквизитов учетной записи;
- изменение привязки учетной записи к группе пользователей;
- удаление учетной записи;
- удаление группы;
- создание новых объектов;
- изменение атрибутов объектов;
- удаление объектов.

В режиме редактора программа позволяет произвести следующие действия по изменению данных системы:

- создание новых объектов;
- изменение атрибутов объектов;
- удаление объектов.

Программа допускает управление одновременно несколькими проектами, при этом можно создавать под каждый из них свою систему управления, не покидая единого интерфейса.

3.2 Используемые методы

Для обработки и защиты информации ограниченного доступа в программе Simbix.Интеграционный портал применяются следующие программные и организационные методы:

- аутентификация и авторизация пользователей;

Име. № подл.	
Подпись и дата	
Взам. име. №	
Име. №	
Подпись и дата	

					SIMBIX.00003-01 13	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

- разграничение прав пользователей при добавлении, редактировании и удалении данных;
- регистрация в системном журнале последнего изменения данных по следующим параметрам: дата/время, объект, пользователь;
- проверки целостности системы.

Для реализации бизнес-логики пользователя в программе используются следующие программные методы:

- метод «s2CopyNode» для копирования узла и ветки в другого родителя;
- метод «s2CountNodes» для возврата числа выбранных узлов;
- метод «s2DeleteNode» для удаления узла и ветки;
- метод «s2Fetch» для вызова вложенного метода;
- метод «s2GetBranch» для возврата упорядоченного набора потомков (ветки) узла;
- метод «s2GetChildren» для возврата дочерних узлов;
- метод «s2GetCurrentUser» для возврата текущего пользователя;
- метод «s2GetKnee» для возврата набора потомков (ветки) узла;
- метод «s2GetNode» для возврата узла;
- метод «s2Chmod» для изменения прав узла;
- метод «s2Chown» для изменения владельца и/или группы-владельца узла;
- метод «s2GetNodeProto» для возврата прототипа узла;
- метод «s2GetNodes» для возврата узла;
- метод «s2GetParent» для возврата родительского узла;
- метод «s2GetPath» для возврата пути в дереве к узлу;
- метод «s2Hash» для возврата необратимого хэша от строки;
- метод «s2Hex2Str» для декодирования строки из шестнадцатеричного представления;
- метод «s2ImageResize» для уменьшения изображения;
- метод «s2IndexArray» для объявления ключами массива значений полей;
- метод «s2InsertNode» для создания новых узлов;
- метод «s2Login» для авторизации пользователя в системе;
- метод «s2Logout» для прекращения работы текущего пользователя;
- метод «s2MoveNode» для смены родителя узла и ветки;
- метод «s2Php2Js» для конвертации переменной PHP в выражение JSON;
- метод «s2SetOrder» для установления порядка выборки дочерних узлов;
- метод «s2SortNodes» для возврата отсортированного списка узлов;

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист 9
------	------	----------	---------	------	---------------------------	-----------

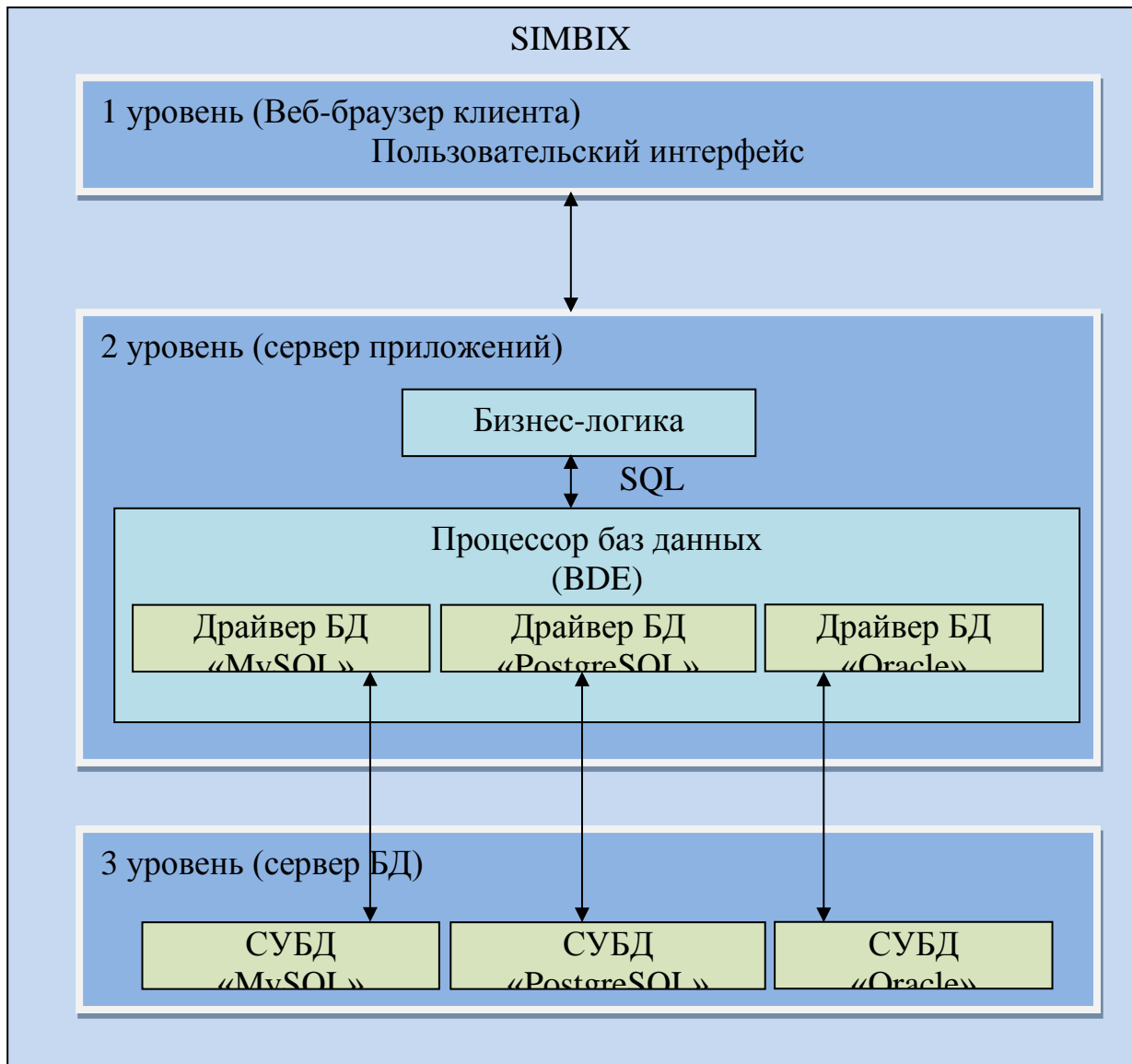


Рисунок 1. Архитектура программы Simbix.

Для работы с программой предусматривается три группы пользователей – администраторы, редакторы и клиенты. Доступ к программе и отнесение пользователей к различным группам осуществляются с применением процедур авторизации и аутентификации пользователя.

Для работы пользователей уровня администратора и редактора используется административный интерфейс, который запускается с использованием веб-браузера на удаленном компьютере.

Администраторы используют функционал для настройки (конфигурирования) программы, редакторы используют функционал управления бизнес-логикой программы и ввода информации, клиенты используют исключительно интерфейс отображения данных через стандартный тонкий клиент (веб-браузер), не имея доступа к администраторским функциям.

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата

- [string "met" = false] Название метода
- [string|array "atr" = false] Атрибуты вызова метода
- [string "var"] Имя переменной для результата в шаблоне

3.4.5 Функция «s2GetBranch»

Функция предназначена для возврата упорядоченного набора потомков (ветки) узла.

Точка входа в функцию – S2_Nodes::getBranch().

Входная и выходная информация:

- [int "NodId" = 1] Номер родительского узла
- [boolean "inclusive" = false] Включать ли узел "NodId" в выборку
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [mixed "sort" = null] Запрос на поля сортировки результата
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

3.4.6 Функция «s2GetChildren»

Функция предназначена для возврата дочерних узлов. Точка входа в функцию – S2_Nodes::getChildren().

Входная и выходная информация:

- [int "NodId" = 1] Номер родительского узла
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [mixed "sort" = null] Запрос на поля сортировки результата
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "limit" = null] Ограничение на количество выбираемых узлов
- [int "offset" = null] Смещение выборки узлов
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

Име. № подл.	Подпись и дата	Взам. инв. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						13

3.4.7 Функция «s2GetCurrentUser»

Функция предназначена для возврата текущего пользователя. Точка входа в функцию – S2_User::getUser().

Входная и выходная информация:

- [string "var"] Имя переменной для результата в шаблоне

3.4.8 Функция «s2GetKnee»

Функция предназначена для возврата набора потомков (ветки) узла. Точка входа в функцию – S2_Nodes::getKnee().

Входная и выходная информация:

- [int "NodId" = 1] Номер родительского узла
- [boolean "inclusive" = false] Включать ли узел "NodId" в выборку
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [mixed "sort" = null] Запрос на поля сортировки результата
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "limit" = null] Ограничение на количество выбираемых узлов
- [int "offset" = null] Смещение выборки узлов
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

3.4.9 Функция «s2GetNode»

Функция предназначена для возврата узла. Точка входа в функцию – S2_Nodes::getNode().

Входная и выходная информация:

- [int "NodId" = 1] Номер узла
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						14

3.4.10 Функция «s2Chmod»

Функция предназначена для изменения прав узла. Точка входа в функцию – S2_Nodes::changeMode().

Входная и выходная информация:

- [int "NodId" = 0] Номер узла (корень ветки)
- [int "NPerm"] Битовая маска
- [boolean "recursive" = false] Признак рекурсивности
- [string "var"] Имя переменной для результата в шаблоне

3.4.11 Функция «s2Chown»

Функция предназначена для изменения владельца и/или группы-владельца узла. Точка входа в функцию – S2_Nodes::changeOwner().

Входная и выходная информация:

- [int "NodId" = 0] Номер узла (корень ветки)
- [int "Owner"] Номер владельца
- [int "Gwner"] Номер группы-владельца
- [boolean "recursive" = false] Признак рекурсивности
- [string "var"] Имя переменной для результата в шаблоне

3.4.12 Функция «s2GetNodeProto»

Функция предназначена для возврата прототипа узла. Точка входа в функцию – S2_Nodes::getNodeProto().

Входная и выходная информация:

- [string "NType" = null] Тип узла
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [string "var"] Имя переменной для результата в шаблоне

3.4.13 Функция «s2GetNodes»

Функция предназначена для возврата узла. Точка входа в функцию – S2_Nodes::getNodes().

Входная и выходная информация:

- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						15

- [mixed "sort" = null] Запрос на поля сортировки результата
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "limit" = null] Ограничение на количество выбираемых узлов
- [int "offset" = null] Смещение выборки узлов
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

3.4.14 Функция «s2GetParent»

Функция предназначена для возврата родительского узла. Точка входа в функцию – S2_Nodes::getParent().

Входная и выходная информация:

- [int "NodId" = 1] Номер узла
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

3.4.15 Функция «s2GetPath»

Функция предназначена для возврата пути в дереве к узлу. Точка входа в функцию – S2_Nodes::getPath().

Входная и выходная информация:

- [int "NodId" = 1] Номер узла
- [int "rootId" = 1] Номер вышестоящего узла
- [boolean "inclusive" = true] Включать ли узел "NodId" в выборку
- [string "condition" = null] Пользовательское условие
- [mixed "NType" = null] Запрос на типы выбираемых узлов
- [mixed "fields" = 3] Запрос на поля выбираемых узлов
- [mixed "linkDetail" = 3] Запрос на виртуальные поля узлов
- [int "NFlag" = 1] Битовая маска флагов видимости
- [int "linkDeep" = 1] Глубина разворачивания ссылочных полей
- [string "var"] Имя переменной для результата в шаблоне

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						16

3.4.16 Функция «s2Hash»

Функция предназначена для возврата необратимого хэша от строки. Точка входа в функцию – S2_System::Hash().

Входная и выходная информация:

- [mixed "value" = "] Исходная строка
- [string "var"] Имя переменной для результата в шаблоне

3.4.17 Функция «s2Hex2Str»

Функция предназначена для раскодирования строки из шестнадцатеричного представления. Точка входа в функцию – S2_API::hex2string().

Входная и выходная информация:

- [mixed "value" = "] Шестнадцатеричное представление
- [string "var"] Имя переменной для результата в шаблоне

3.4.18 Функция «s2ImageResize»

Функция предназначена для уменьшения изображения. Точка входа в функцию – S2_API::imageResize().

Входная и выходная информация:

- [array "image"] Массив, описывающий изображение
- [int "size"] Размер, до которого нужно уменьшить картинку
- [int "dimension" = 1] Сторона, по которой уменьшение
- [array "types" = array().] Список допустимых типов изображения
- [int "quality" = 80] Качество создаваемого изображения
- [string "var"] Имя переменной для результата в шаблоне

3.4.19 Функция «s2IndexArray»

Функция предназначена для объявления ключами массива значений полей. Точка входа в функцию – S2_System::indexArray().

Входная и выходная информация:

- [array "array"] Исходный массив
- [string "key"] Название поля, значения которого станут ключами
- [string "var"] Имя переменной для результата в шаблоне

3.4.20 Функция «s2InsertNode»

Функция предназначена для создания новых узлов. Точка входа в функцию – S2_Nodes::insertNode().

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						17

Входная и выходная информация:

- array \$params: Описание узла
- [string "var"] Имя переменной для результата в шаблоне

3.4.21 Функция «s2Login»

Функция предназначена для авторизации пользователя в системе. Точка входа в функцию – S2_User::Login().

Входная и выходная информация:

- [string "Login" = false] Учетная запись пользователя
- [string "Password" = false] Пароль пользователя
- [string "var"] Имя переменной для результата в шаблоне

3.4.22 Функция «s2Logout»

Функция предназначена для прекращения работы текущего пользователя. Точка входа в функцию – S2_User::Logout().

Входная и выходная информация:

- [boolean "Nobody" = false] Выйти сразу в "Nobody" (без стека)
- [string "var"] Имя переменной для результата в шаблоне

3.4.23 Функция «s2MoveNode»

Функция предназначена для смены родителя узла и ветки. Точка входа в функцию – S2_Nodes::moveNode().

Входная и выходная информация:

- [int "NodId" = 1] Номер копируемого узла (корень ветки)
- [int "ParId" = 1] Номер целевого (нового) родителя узла
- [string "var"] Имя переменной для результата в шаблоне

3.4.24 Функция «s2Php2Js»

Функция предназначена для конвертации переменной PHP в выражение JSON. Точка входа в функцию – S2_API::php2js().

Входная и выходная информация:

- [mixed "value" = false] Переменная PHP
- [string "var"] Имя переменной для результата в шаблоне

3.4.25 Функция «s2SetOrder»

Функция предназначена для установления порядка выборки дочерних узлов. Точка входа в функцию – S2_Nodes::setOrder().

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						18

Входная и выходная информация:

- [int "NodId" = 1] Номер родительского узла
- [string "sortBy" = false] Сортировать детей по ...
- [array "children" = null] Список номеров узлов в требуемом порядке
- [string "var"] Имя переменной для результата в шаблоне

3.4.26 Функция «s2SortNodes»

Функция предназначена для возврата отсортированного списка узлов. Точка входа в функцию – S2_API::sortNodes().

Входная и выходная информация:

- [array "nodes" = array().] Список узлов
- [array "sort" = array().] Поля сортировки результата (см. sortNodes)
- [string "var"] Имя переменной для результата в шаблоне

3.4.27 Функция «s2Str2Hex»

Функция предназначена для кодировки строки в шестнадцатеричное представление. Точка входа в функцию – S2_API::string2hex().

Входная и выходная информация:

- [mixed "value" = "] Исходная строка
- [string "var"] Имя переменной для результата в шаблоне

3.4.28 Функция «s2UpdateNode»

Функция предназначена для изменения узла. Точка входа в функцию – S2_Nodes::updateNode().

Входная и выходная информация:

- array \$params: Описание узла
- [string "var"] Имя переменной для результата в шаблоне

3.4.29 Функция «append»

Функция предназначена для создания массива или добавления элемента массива в шаблоне.

Входная и выходная информация:

- [array "from" = array().] Исходный массив
- [string "key" = 0] Назначаемый ключ массива
- [string "value" = "] Добавляемое значение
- [string "var"] Имя переменной для результата в шаблоне

Име. № подл.	Подпись и дата
Взам. име. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 13	Лист
						19

3.4.30 Функция «explode»

Функция предназначена для разбивки строки на массив в шаблоне.

Входная и выходная информация:

- [string "values" = "] Строка со списком значений
- [string "delimiter" = ','] Разделитель значений
- [string "var"] Имя переменной для результата в шаблоне

3.5 Связи программы с другими программами

Программа Simbix.Интеграционный портал представляет собой клиент-серверное веб-приложение, построенное на принципах трехуровневой архитектуры, описание которой приведено в разделе 3.3 настоящего документа.

Второй и третий программный уровень реализуется на серверах, запросы к которым из пользовательского интерфейса (первый программный уровень) поступают по каналам internet (intranet) с использованием большинства современных веб-браузеров.

Для работы программы Simbix.Интеграционный портал на компьютерах клиента и сервере должны использоваться лицензионные программы, перечень, наименование и версии которых приведены в разделе 1.2 настоящего документа.

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. №	Подпись и дата	SIMBIX.00003-01 13				Лист
									20
Изм.	Лист	№ докум.	Подпись	Дата					

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Программа Simbix может работать на однопроцессорных и многопроцессорных компьютерах, построенных на платформе Intel x86.

Технические характеристики сервера должны позволять устанавливать следующие программные средства:

- одна из операционных систем UNIX/Linux, Microsoft Windows 2000 (Professional, Server, Advanced Server), Windows Server 2003, Windows Server 2008;
- веб сервер: Apache 2/Nginx с поддержкой PHP5 и mod_rewrite;
- один из серверов СУБД: MySQL 5.0 и выше, PostgreSQL 8.3 и выше, Oracle 9.2.

Технические характеристики компьютера клиента должны позволять устанавливать следующие программные средства:

- операционная система – Windows, в версии Windows XP или более поздних версиях;
- интернет-обозреватель – Microsoft Internet Explorer (не ниже версии 6.0), Mozilla Firefox (не ниже версии 3.5), Опера (не ниже версии 9.64 RU) или аналогичный.

Рекомендуемые характеристики компьютеров для работы программы Simbix.Интеграционный портал приведены в таблице 1.

Таблица 1. Рекомендуемые характеристики компьютеров

Наименование	Техническая спецификация номенклатуры программно-технических средств	Кол-во
Сервер	<p>Технические характеристики сервера должны быть не хуже характеристик, рекомендуемых для развертывания используемой версии СУБД.</p> <p>Минимальные характеристики:</p> <ul style="list-style-type: none"> • процессор – 64-разрядный процессор с таковой частотой не менее 2 ГГц; • оперативная память - не менее, 2ГБ; • жесткий диск - свободное место не менее 10 Gb; • сетевая карта Ethernet 10/100/1000 Base-T – 1 шт.; • устройство чтения компакт-дисков типа CD/DVD 	1
Компьютер клиента	<p>Минимальные характеристики:</p> <ul style="list-style-type: none"> • процессор - 2400 МГц; • оперативная память – 512 Мбайт; • монитор от 15", разрешение не менее 1024x768; • жесткий диск - свободное место не менее 500 Mb; • русскоязычная клавиатура; • интерфейсы подключения PS/2 и USB; • сетевая карта Ethernet 10/100/1000 Base-T – 1 шт.; • устройство чтения компакт-дисков типа CD/DVD; • графический манипулятор (мышь) 	1

Ине. № подл.	Подпись и дата
Взам. ине. №	Ине. №
Подпись и дата	Подпись и дата

5 ВЫЗОВ И ЗАГРУЗКА

Загрузка (инсталляция) программы Simbix.Интеграционный портал выполняется на сервер приложений с помощью встроенного в комплект поставки установщика с графическим интерфейсом. Процесс установки подробно описан в руководстве администратора.

Программа Simbix.Интеграционный портал запускается на сервере приложений веб-сервером «Apache» при дистанционном запросе на выполнение требуемой функции из административного интерфейса клиента. После выполнения запроса программа завершает свою работу.

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. №	Подпись и дата	SIMBIX.00003-01 13					Лист
										22
Изм.	Лист	№ докум.	Подпись	Дата						

