Для добавления нового движка шаблонизации на сайт в файл */bitrix/php\_interface/init.php* необходимо добавить следующее:

1. Глобальную переменную *$arCustomTemplateEngines*, которая содержит ассоциативный массив, каждый элемент которого имеет вид:
*"код\_шаблонизатора" => array(
      "templateExt" => array("расширение1"[, "расширение2"...]),
      "function" => "имя\_функции\_подключения\_движка"
   )*
где:
*"код\_шаблонизатора"* - произвольное уникальное в рамках сайта слово;
*"расширениеN"* - расширение файла, который должен обрабатываться этим движком шаблонизации;
*"имя\_функции\_подключения\_движка"* - имя функции, которая будет вызываться, если шаблон компонента имеет указанное расширение.

2. Функцию подключения движков:
*function имя\_функции\_подключения\_движка($templateFile, $arResult, $arParams, $arLangMessages, $templateFolder, $parentTemplateFolder, $template),*
где:
     $templateFile – путь к файлу шаблона относительно корня сайта,      $arResult – массив результатов работы компонента,
     $arParams – массив входных параметров компонента,
     $arLangMessages – массив языковых сообщений (переводов) шаблона,
     $templateFolder – путь к папке шаблона относительно корня сайта (если шаблон лежит не в
папке, то эта переменная пуста),
     $parentTemplateFolder - путь относительно корня сайта к папке шаблона комплексного
компонента, в составе которого подключается данный компонент (если компонент
подключается самостоятельно, то эта переменная пуста),
     $template – объект шаблона.

Рассмотрим подключение движков на конкретных примерах.



**Пример подключения движка Smarty:**

В массиве *$arCustomTemplateEngines* регистрируется движок Smarty:

*global $arCustomTemplateEngines;
$arCustomTemplateEngines = array(
   "smarty" => array(
      "templateExt" => array("tpl"),
      "function" => "SmartyEngine"
   ),
);*

В функции *SmartyEngine* инициализируются параметры движка в соответствии с требованиями Smarty (см. систему помощи Smarty). Далее в Smarty передаются переменные результатов работы компонента, входных параметров, языковых сообщений и т.д., а в конце вызывается метод обработки и показа шаблона Smarty:
 *function SmartyEngine($templateFile, $arResult, $arParams, $arLangMessages, $templateFolder, $parentTemplateFolder, $template)
{
   if (!defined("SMARTY\_DIR"))
      define("SMARTY\_DIR", "<абсолютныйпутькдвижку Smarty>/libs/");

   require\_once( '<абсолютныйпутькдвижку Smarty>/libs/Smarty.class.php' );

   $smarty = new Smarty;

   $smarty->compile\_dir = "<абсолютныйпутькдвижку Smarty>/templates\_c/";
   $smarty->config\_dir = "<абсолютныйпутькдвижку Smarty>/configs/";
   $smarty->template\_dir = "<абсолютныйпутькдвижку Smarty>/templates/";
   $smarty->cache\_dir = "<абсолютныйпутькдвижку Smarty>/cache/";

   $smarty->compile\_check = true;
   $smarty->debugging = false;

   $smarty->assign("arResult", $arResult);
   $smarty->assign("arParams", $arParams);
   $smarty->assign("MESS", $arLangMessages);
   $smarty->assign("templateFolder", $templateFolder);
   $smarty->assign("parentTemplateFolder", $parentTemplateFolder);

   $smarty->display( $\_SERVER["DOCUMENT\_ROOT"].$templateFile );
}*
В строке *"<абсолютныйпутькдвижку Smarty>"* указывается абсолютный путь к движку Smarty.



**Пример подключения движка XML/XSLT:**Сначала регистрируем движок:

*global $arCustomTemplateEngines;
$arCustomTemplateEngines = array(
   "xslt" => array(
      "templateExt" => array("xsl"),
      "function" => "XSLTEngine"
   ),
);*

Функция инициализации параметров движка:

*function CreateXMLFromArray($xDoc, $xNode, $ar)
{
   foreach($ar as $key=>$val)
   {
      if(!is\_string($key) || strlen($key)<=0)
         $key = "value";*