

ASTS Bridge+ (пре-трейд контроль)

Внимание: так как при использовании функционала пре-трейд контроля поступающие в торговую клиринговую систему Московской Биржи транзакции проходят дополнительную обработку в компоненте, не являющимся частью шлюза ASTS Bridge, это может увеличить время постановки/снятия заявок. Установка на шлюзовом сервере стороннего ПО (в данном случае – модуля пре-трейд контроля) осуществляется участником на свой страх и риск. В случае возникновения проблем в данном ПО, в технической поддержке со стороны Биржи может быть отказано.

Данный документ описывает функционал шлюза, позволяющий подключать к нему внешние DLL для выполнения пре-трейд контроля. Внешняя DLL вызывается каждый раз при исполнении клиентской транзакции, и может либо разрешить, либо запретить ее выполнение.

Внешняя DLL должна реализовывать три функции:

```
const
    ORDER_VALIDATION_ERROR_LEN = 256;
```

Delphi

```
function UserConnected(UserId, ErrorMessage: PAnsiChar): BOOL; stdcall;
```

C++

```
BOOL WINAPI UserConnected(char *UserId, char* ErrorMessage);
```

Эта функция вызывается при подключении клиента. На вход ей подается идентификатор пользователя. Функция должна вернуть TRUE, если она разрешает подключиться данному пользователю, иначе – FALSE. В случае запрета подключения, функция может вернуть текстовую строку длиной до 255 символов с указанием причины в параметре ErrorMessage. При запрете подключения клиентское приложение получит ошибку -30 (MTE_LOGON).

Delphi

```
procedure UserDisconnected(UserId: PAnsiChar); stdcall;
```

C++

```
void WINAPI UserDisconnected(char * UserId);
```

Эта функция вызывается при отключении клиента от шлюза. На вход ей подается идентификатор пользователя. Эта функция не возвращает никаких значений.

Delphi

```
function ValidateOrderEx(UserId, TransName, Params, JsonParams,
    ErrorMessage: PAnsiChar): BOOL; stdcall;
```

C++

```
BOOL WINAPI ValidateOrderEx(char * UserId, char *TransName,
    char* Params, char* JsonParams, char *ErrorMessage);
```

Эта функция вызывается при каждом исполнении транзакции клиентом. На вход ей подаются идентификатор пользователя (*UserId*), идентификатор транзакции (*TransName*) и параметры транзакции:

- в виде буфера Торговой системы (*Params*);
- в формате JSON (*JsonParams*). JSON формируется шлюзом на основе описания формата транзакции, указанной в параметре *TransName*. Если клиент шлюза пытается передать недопустимое название транзакции, или передаваемые параметры не соответствуют формату транзакции, *JsonParams* будет содержать пустой указатель (NULL).

Функция должна вернуть TRUE, если она разрешает исполнение транзакции, иначе – FALSE. В случае запрета исполнения транзакции, функция может вернуть текстовую строку длиной до 255 символов с указанием причины в параметре *ErrorMsg*. При запрете транзакции клиентское приложение получит ошибку -18 (MTE_TRANSREJECTED).

Примечание. В целях совместимости поддерживается также старый прототип функции валидации:

Delphi

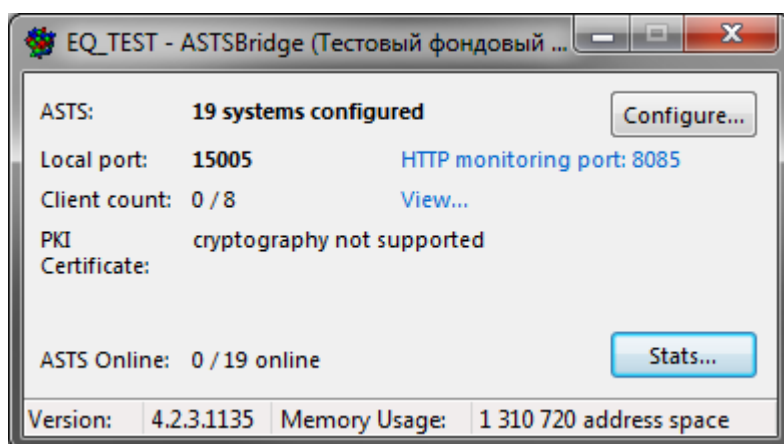
```
function ValidateOrder(UserId, TransName, Params,  
    ErrorMsg: PAnsiChar): BOOL; stdcall;
```

C++

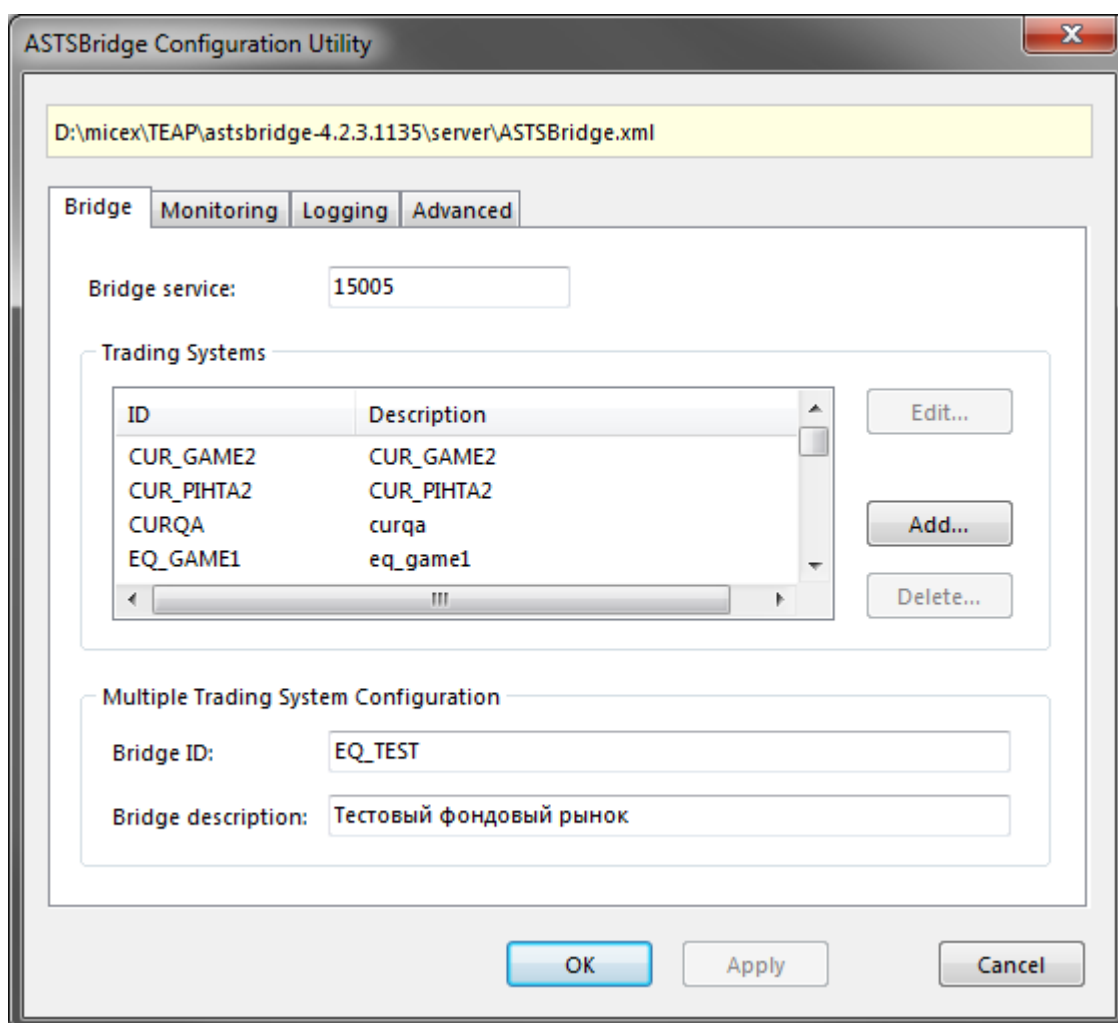
```
BOOL WINAPI ValidateOrder(char * UserId, char *TransName,  
    char* Params, char *ErrorMsg);
```

Пример простейшей DLL и ее исходный код находится в файле *OrderValidation.dpr*. Эта демонстрационная DLL на каждую проходящую через нее транзакцию задает вопрос «Разрешить или запретить эту транзакцию». Можно подключить эту DLL к какому-нибудь идентификатору пользователя (см. ниже), поставить заявки через тестовый клиент *TEClient.exe* (см. дистрибутив шлюза) и увидеть, как работает эта схема.

Для каждого пользователя может быть задана своя внешняя DLL. Настройка шлюза, начиная с версии 4.1, реализована через графический интерфейс. Запустите шлюз и нажмите кнопку «Configure...»:

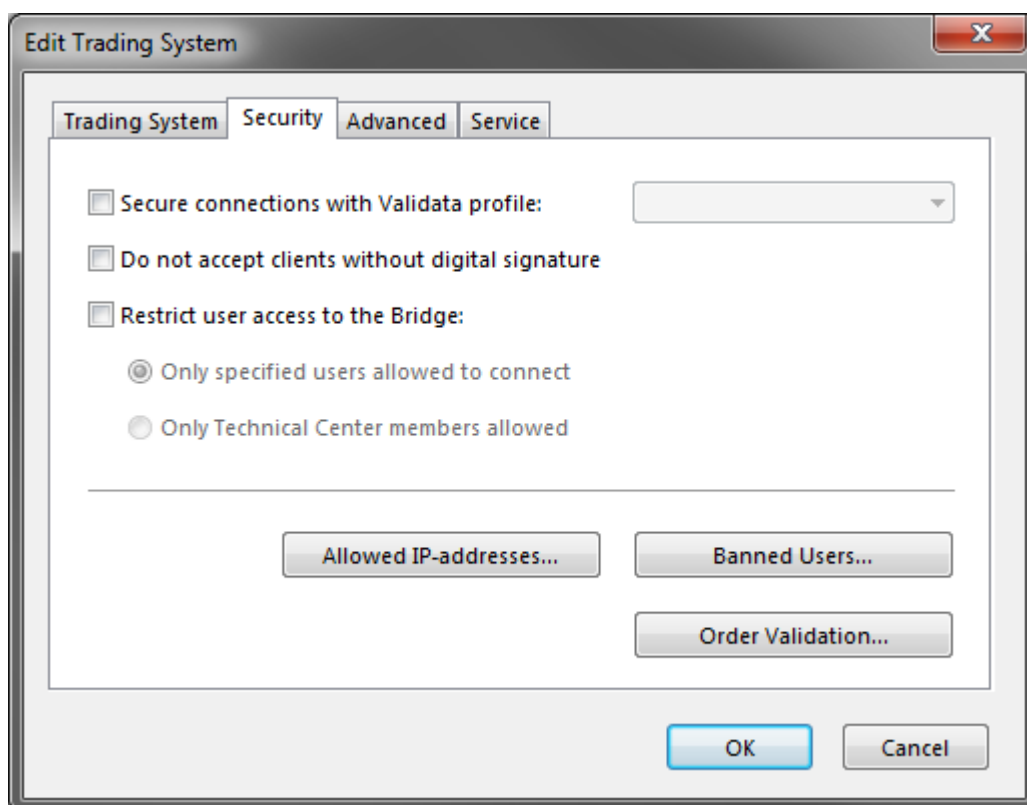


Запустится утилита редактирования конфигурационного файла:

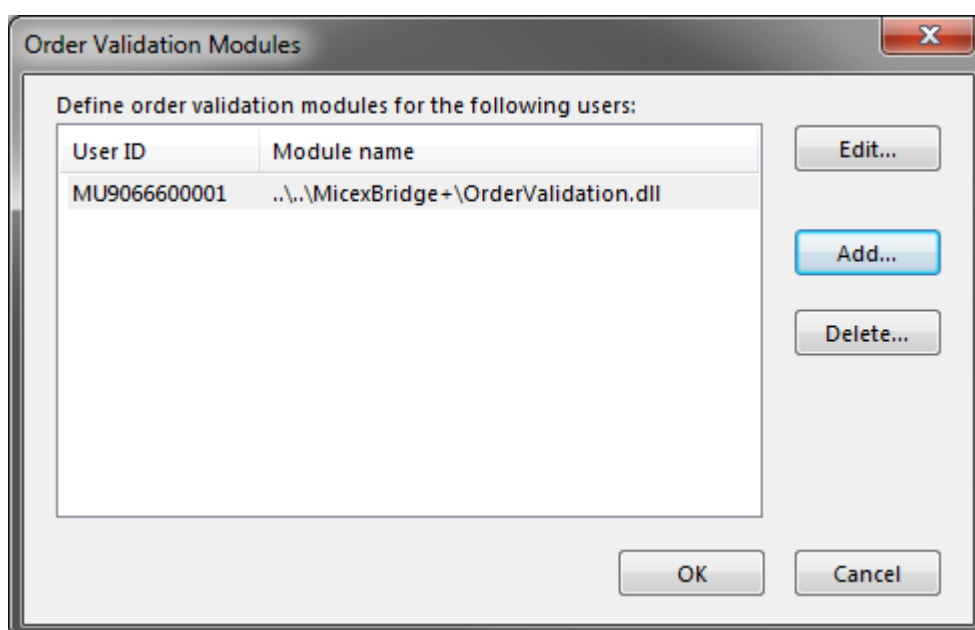


Чтобы получить доступ к новым возможностям шлюза, необходимо использовать XML-формат хранения настроек (нажмите кнопку «Convert to XML» для конвертирования настроек из старого формата в новый).

После этого необходимо открыть окно настройки подключения к торговой системе (кнопка «Edit»):



На вкладке «Security» нажать кнопку «OrderValidation». Откроется окно для подключения внешних DLL:



В этом окне для каждого пользователя можно указать DLL, которая будет загружаться при подключении данного пользователя, и разрешать/запрещать его транзакции.