

Memo

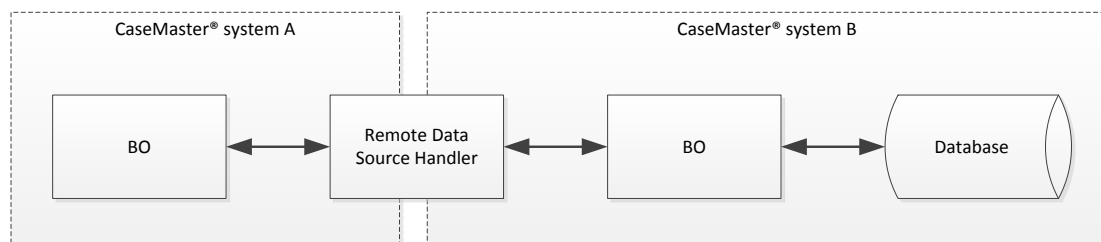
9 K N O T S

By	Bertus Dispa	
Regarding	CaseMaster Remote Data Source Handler	
To	Team	
Date	22MAR15	
Status	In commercial confidence; distribution to relevant project team members and selected 3 rd parties allowed	
History	V1.0, 22MAR15	BD

Synopsis

This document describes how to configure and use the Remote data source handler. The Remote data source handler is designed for easy access to remote data.

CaseMaster® System A can use a BO as if the data exists locally; the data however sits in a database managed by CaseMaster® System B. System A acts as the Remote Client; System B as the Remote Server. The Remote data source handler communicates using SOAP requests.



Some examples of where the Remote data source can be useful:

- A feature is added to CaseMaster® systems used by clients that has basic access to the help desk system of the CaseMaster® reseller
- A healthcare system has very strict data protection requirements and cannot store patient identifying information (name, date of birth, hospital patient reference) on the same server as medical data

- Two CaseMaster® systems need to have an interface to exchange data
- Etc

Compared to the Alternatives

- *Connecting as alternative data source.* This will only work when the client and server system are hosted on the same server (or see next alternative)
- *Remote database connection.* This is often not allowed / possible; a remote database connection is difficult for some RDBMS-es (e.g. Jet / Access) and maybe undesirable because of security reasons
- *Web service.* Means you have to develop client and server logic; client does not work as a data source

Configuring the Remote Client

The main step is to add the data source handler to the `incDB.xml` file of the client system.

```
<datasource type="channel" name="remote" class="oCMDS.clsDSHRemote">
  <server>https://77.246.171.248/cmPDV.api</server>
  <login>admin</login>
  <password>adminPa$$w0rt</password>
  <encryptionKey>enCrypt10n</encryptionKey>
  <BOSpecifics>
    <BO>
      <localName>pdv/patient</localName>
      <remoteName>pdv/systems/lsms/patient</remoteName>
      <remoteAutomatics>Y</remoteAutomatics>
      <localAuditTimestamp>N</localAuditTimestamp>
      <localAuditUser>N</localAuditUser>
    </BO>
  </BOSpecifics>
</datasource>
```

(Note that you can add multiple instance of the Remote data source handler to a client system; each with its' own settings).

Tag	Notes
server	Mandatory; the URL of the server system

login	Mandatory; User-id of a valid login on the server system
password	Mandatory; Password for the user-id
encryptyKey	Optional; key used to encrypt data; if omitted, the data with the server is not encrypted
BOSpecifics	Optional; section can contain entries for additional control on dealing with specific business objects. See paragraph on 'BO Specifics'

Needless to say that for test / development purposes you can have the server URL point to 127.0.0.1.

Configuring the Remote Server

The remote server needs little configuration; the file `cmAsp/DSHRemote.asp` needs to be present and the following tags need to be included in the application configuration file.

```

<DSHRemote>
  <loginUser>remoteXS</loginUser>
  <password>any123</password>
  <encryptionKey>enCrypT10n</encryptionKey>
</DSHRemote>

```

The user details must be a valid user-id / password combination for a user on the server system. This user-id needs only basic access rights as it is only used to kickstart the server-side logic for the Remote data source handler.

The encryption key (when used, is optional) must be the same as set in the `incDB.xml` file of the client system(s).

A Typical Set-up

The following is an example of a typical set-up where 'HealthCare' is the client system and 'PatientDataVault' is the server system. The server system is used to store the name, date of birth, patient reference and primary key of patient data which, for data protection reasons, cannot be stored system 'HealthCare'.

System	Component	Use
--------	-----------	-----

HealthCare	PDV data source	Set up as Remote data source handler (PDV is short for 'Patient Data Vault')
HealthCare	Patient BO	BO with all medical data; simply linked to the primary data source
HealthCare	PDV BO	BO linked to the Remote data source handler; contains PK, name, date of birth and patient reference
PatientDataVault	PDV data source	Set up as an alternative data source but has exact same data source name as the Remote data source on the client system
PatientDataVault	PDV BO	Exact copy of the PDV BO from the client system but is now linked to an alternative relation database data source
HealthCare	Event action on insert on Patient BO	Copies primary key, name, date of birth and patient reference to an instance of the PDV BO and insert that. This scenario assumes that the name, date of birth and patient reference are memory-only attributes which are only used to prompt the user for the data but are never stored locally
HealthCare	Event action on delete on Patient BO	Ensures that the data in the Patient Data Vault is deleted when the patients' medical record is deleted

We use the same data source name on the client- and server system (but one being of type Remote and the other Alternative so we can safely share the same BO between the two systems. This makes it easier to keep the two systems in sync.

Using BO Specifics

The section 'BOSpecifics' in the `incDB.xml` file of the client system is optional but can be used to get a bit more control over the Remote behaviour for different business objects.

Tag	Notes
<code>localName</code>	Mandatory; the name of the BO on the client system
<code>remoteName</code>	The name of the BO on the server system; assumes the same name as on the client system when omitted
<code>remoteAutomatics</code>	Defaults to false; only relevant where the BO has a primary key of type 'automatic'. Setting this tag to 'true' means that the BO will be assigned a new automatic on the server (using the sequence of the server)
<code>localAuditTimeStamp</code>	Defaults to false; only relevant for business objects that have auditing set. Setting this tag to 'true' means that the timestamp of the client system will be used for <code>zXCrtDWhn</code> and <code>zXUpdtDWhn</code> . This can be useful when the timestamp is important to the client and an optional time difference between server and client would cause problems
<code>localAuditUser</code>	Defaults to false; only relevant for business objects that have auditing set. Setting this tag to 'true' means that the <code>zXCrtDBy</code> and <code>zXUpdtDBy</code> attributes will be set to the current user; note that this user-id may not exist on the server system. By default the user-id as specified in the <code>incDB.xml</code> file of the client system will be used (and this user <i>must</i> exist on the server system)

Typical example:

```
<BOSpecifics>
  <BO>
    <localName>pdv/patient</localName>
    <remoteName>pdv/systems/lsms/patient</remoteName>
    <remoteAutomatics>Y</remoteAutomatics>
    <localAuditTimeStamp>N</localAuditTimeStamp>
    <localAuditUser>N</localAuditUser>
```

```
</BO>
</BOSpecifics>
```

Security Best Practices

- On the server
 - Do not 'simply add' server support for Remote to an existing system but create a mini-system specifically for this purpose
 - Ensure that the user-id as specified in the 'DSHRemote' section of the `inczX.xml` file only has minimum rights
 - Ensure that the dedicated application only supports https
 - Ensure that the user-id as specified in the `incDB.xml` file of the client(s) has minimal rights
 - Consider using the security section of the BO's that is subject to the Remote data source to restrict the actions allowed
 - Set an encryption key
- On the client
 - Use `https://` as the protocol
 - Set an encryption key
 - Consider restricting access to the BO by using vaults or user groups on the security section (even if this means you cannot share the same BO on client and server system)

Advanced Use

Cannot use Same Folder on Client and Server

It may be possible that the client- and server system cannot use / do not want to use the same BO folder structure.

For this scenario you can use the 'BOSpecific' section and add an entry using a different 'localName' and 'remoteName'.

You may still be able to use the same BO file (even though, on one of the systems the name differs from the path) but this may have some minor consequences (you can no longer point to this BO in a foreign key, some features of the editor may no longer work).

Using Event Actions

You can use event actions in a BO which is subject to the Remote data source. If you want to use the same BO file on the client and server, you may want to add a condition to each event action to ensure it only fires on the client or server (e.g. by referring to an entry in the configuration file or so).

Setting Access Security

It is advisable to use the security section of the BO to restrict access where possible. Do appreciate that the groups or vaults used in the security sections are interpreted on client- and server system when you choose to use the same BO file on both systems.

Using Foreign Keys

You can use foreign keys in BO's subjected to the Remote data source as you can have other BO's have a FK to one. Do however appreciate that the FK will be resolved on both client- and server system and that the target BO must thus exist on both as well.

Restrictions

Performance

The use of the Remote data source is pretty transparent to the developer. It is important to appreciate that each request can result in up to 4 messages being send back-and-forth between client and server.

Atomic Transactions

The server is not part of the transaction on the client. Imagine a scenario where the server call succeeds and the client fails after the Remote call and does a rollback. The data change on the server will not be rolled back.

VB Debugging

It is very hard to use VB debugging when both client and server exist on the local machine. You can debug client or server but both at the same time.

- *Debug client.* Force a remote call not using the debugger; this will ensure that the DLL's are in memory associated with the IIS processes. Now start the debugger and test using the LB / method test facility from the editor

- *Debug server.* Start the editor so it loads the DLL's. Now do a IIS reset and start the debug environment; you can now use the LB / method test facility of the editor to force a Remote server call

<End of document>