

MERMAID

EU Project No: IST-1999-10637

**Workpackage 7: Broker Applications Prototyping
Example Application for
Marine Emergency Response**

Deliverable No: D9.1

(Rev 1)

June 2002



Report Title:	MERMAID – WP7: Example Application for Marine Emergency Response
Customer:	European Commission, Directorate-General Information Society, IST Programme
BMT Report no:	13300/D/9.1
Deliverable nos:	D9.1
Report status:	Rev 1
Date:	June 2002
Contact details:	<p>BMT MARINE INFORMATION SYSTEMS LTD Grove House, 7 Ocean Way, Ocean Village, Southampton, Hampshire, SO14 3TJ. United Kingdom.</p> <p>Tel: +44 (0) 2380 232222 Fax: +44 (0) 2380 232891</p> <p>e-mail: mis@bmtmis.demon.co.uk Website: http://www.bmtmis.com</p>

	Name	Signature	Date
Author:	Chris Rawlings		
Approved by:	Paul Taylor		

This report is commercial-in-confidence. Any information contained herein should not be communicated to any third party without prior permission of BMT Marine Information Systems Ltd.



Contributors

Company	Name
BMT	Chris Rawlings
	Paul Taylor

Distribution List

Company	Name	Number of Copies
EU Commission (DG-XIII)	Guy Weets	2
External Reviewers	Keith Jeffrey	1
	Bernard Donnay	1
	Wolf-Fritz Riekert	1
BMT	Paul Taylor	1
TXT	Matteo Villa	1
CEDRE	Vincent Gouriou	1
Met. Office	Paul Lancaster	1
IMGW-OM	Wlodek Krzyminski	1
NC	Kees Koning	1



Revision

Revision Number	Date	Author	Purpose of Revision
0	20/06/2002	Chris Rawlings	First Internal Draft
1	08/07/2002	Paul Taylor	Final Release Draft



Table of Contents

1	INTRODUCTION	6
2	SYSTEM OVERVIEW	7
2.1	System Design	7
2.1.1	System Diagram.....	7
2.2	Detailed Design	8
2.2.1	Example Application Components.....	8
2.3	BMTNotify.....	8
2.3.1	STDMETHODIMP IsSignalled(VARIANT_BOOL *pvbSignalled)...	8
2.3.2	STDMETHODIMP FireNotify().....	8
2.4	BMTDatabroker	8
2.5	CBMTDataBrokerApp.....	9
2.6	CInterfaceSheet	9
2.7	CLoginPage.....	9
2.8	CNewSearchPage.....	10
2.9	CSearchPage	10
2.10	CBillingDetailsPage.....	10
2.11	CCreditCardPage.....	10
2.12	CConfirmDeliveryPage.....	10
2.13	CConfirmOrder	11
2.14	CDownloadStatusPage	11
2.15	CDlgViewProduct	11
2.16	CMermaidInterface.....	11
2.16.1	Methods	11
2.17	CMermaidBillingAddress.....	11
2.18	CMermaidPayment.....	11
2.19	CMermaidSearchParams.....	11



1 INTRODUCTION

The overall objective of the MERMAID project was to develop a seamless, minimum intervention link (a Data Broker) to allow end users working in the marine environmental emergency application domain to access and use large distributed datasets of environmental parameters. This Data Broker is primarily accessible to most users as a web site through a standard web browser (Workpackage 6). However, it was also decided to demonstrate that the Data Broker could be accessed directly through third party applications. A generic Interface (IF) component was developed (workpackage 4) that would allow access to the Data Broker via applications developed during Workpackage 7. A simple data viewer application was developed by the Polish partner IMGW (see Deliverable D9.2). In addition, two existing marine emergency response applications (a Search and Rescue system (SARIS) and an Oil Spill system (OSIS) – which are currently used operationally by two of the consortium partners; CEDRE and NC) were modified to allow seamless access to the Data Broker to obtain the necessary hydrodynamic data to drive the models. The user therefore does not need to exit the application in order to access the Data Broker, but can interface with MERMAID from ‘within the application’.

This document details the design of the Example Application for Integration into Marine Emergency Response Applications. This application takes the form of an intuitive Wizard-based application with runtime communication methods to the SARIS and OSIS Emergency response applications allowing online application integration to the Marine Data Broker service via the Interface component.

2 SYSTEM OVERVIEW

The Mermaid application for emergency responses uses Microsoft COM technology to enable the dynamic invocation of the wizard application from within an emergency response application. In this case OSIS. The use of COM as an invocation mechanism allows the search information to be seamlessly passed from the emergency response application to the example application which then ensures that a data search for the correct data theme is then executed. The example application allows the initiation of the download to the host machine and monitors the download status. A signalling object is then used to notify the emergency response application that the download is complete allowing the recovery and import of the downloaded file.

2.1 System Design

2.1.1 System Diagram

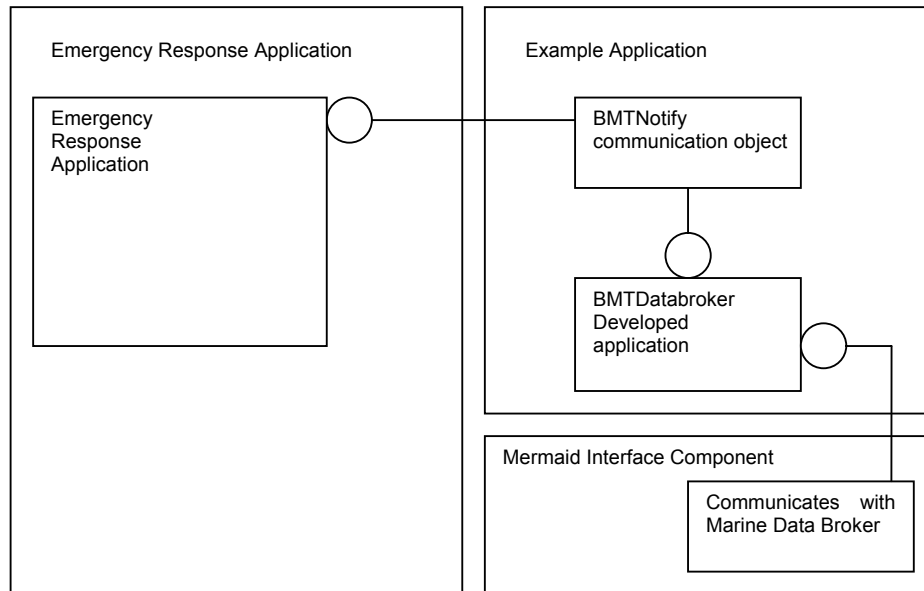


Figure 1. System Design Diagram



2.2 Detailed Design

2.2.1 Example Application Components

The example application components comprise an in-process and out-of-process COM server that work together for the example application. The example application itself; BMTDatabroker.exe can be invoked from the command line or dynamically through an out-of-process server COM invocation. In the case of the latter an exposed proxy interface is used to communicate to the GUI of the application, supplying it with the required search parameters for the dataset search. In this case the invoking application can request an interface to the BMTNotify signalling object and can monitor the status of the object to test for download status. When invoked from the command line the application acts as a simple wizard-based download client providing an alternative to the Web-based download method.

2.3 BMTNotify

The BMTNotify object implements the IBMTNotify interface and is a simple holder of a signalled state object. Process synchronisation is taken care of by the COM marshalling framework to only allow one reference of the interface to affect the object at any one time. As with any COM interface, once an application has finished with the interface, it must be released.

2.3.1 STDMETHODCALLTYPE IsSignalled(VARIANT_BOOL *pvbSignalled)

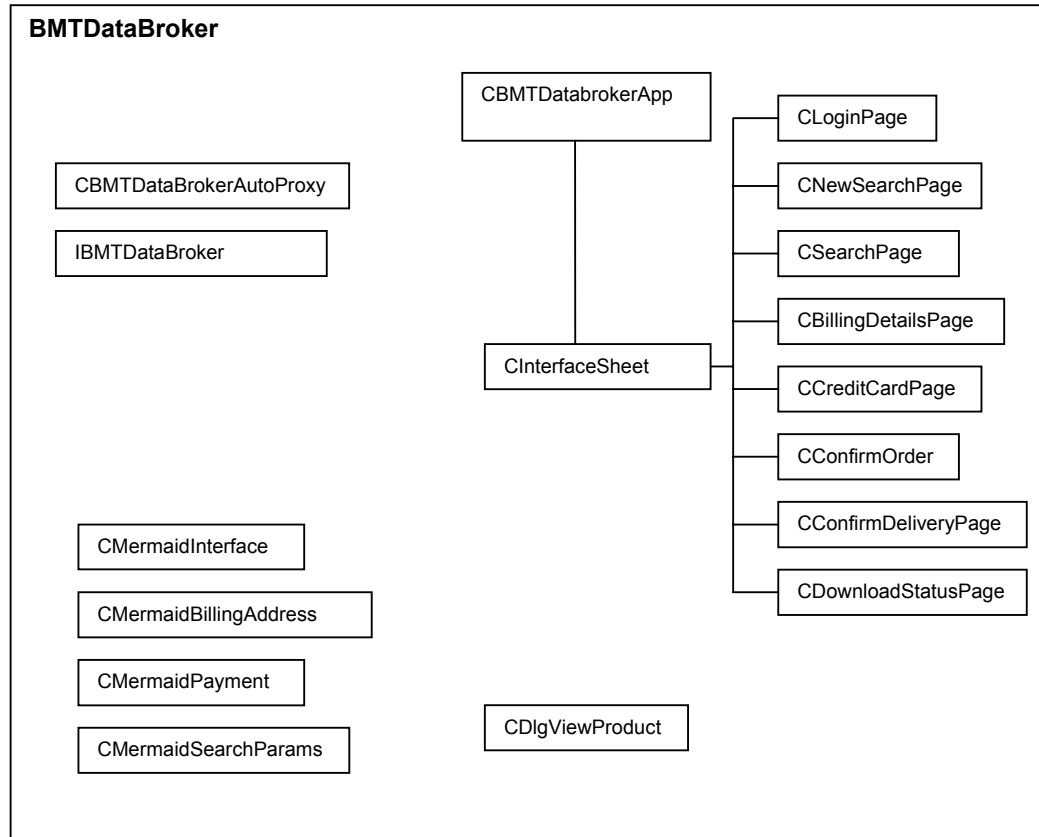
This method is used to test the state of the signalled object. The signalled state of the object is copied into the pvbSignalled parameter, and the success of the method call is returned as an HRESULT.

2.3.2 STDMETHODCALLTYPE FireNotify()

This method sets the signal object in its signalled state. This is irreversible. It is used to notify to an external object holding a reference to this object that the download of any purchased datasets is complete.

2.4 BMTDatabroker

The BMTDatabroker application is a windows application that exposes a hidden COM interface that can be used via standard windows automation to invoke the application. In addition, this interface can 'see' the wizard application and can be used to initialise the application with the search parameters as required. The application itself is a wizard-based application that communicates with the Marine Data Broker through the Interface component. The following class diagram represents the relationship of the classes within the application:



2.5 CBMTDataBrokerApp

This class is the main application class derived from the standard windows CWinApp class provided by MFC. It provides all of the standard functionality expected of the application. It is responsible for creating the CInterfaceSheet class and attaching the wizard pages to it and instantiating the wrapper classes for the Mermaid interface component – CMermaidInterface,

2.6 CInterfaceSheet

This class is derived from the standard MFC CPropertySheet class and is used as a place holder for the wizard pages which contain all of the functionality. The only override performed is the alteration of the background colour of each of the property pages to one standard colour encoded within the CInterfaceSheet class. This is an internal and non-modifiable property.

2.7 CLoginPage

This class is derived from the standard MFC class CPropertyPage and deals entirely with the functionality concerning the logging in to the Data Broker from the client machine. The user is presented with fields to enter his registered Mermaid User Name and password. Pressing the 'Next' button will initiate the login process. Continuation throughout the wizard is only allowed if logon is



successful. If logon fails, the information returned by the Mermaid interface component is returned to the user in a message box.

2.8 CNewSearchPage

This class is derived from the standard MFC class CPropertyPage. This page offers the ability to set 3 search parameters – the GeoSpatial coordinates of the search, the temporal coordinates of the search and the shape of the search. The additional parameters required by the mermaid interface component to initiate a search are coded within the BMTDatabroker application. When the user presses the ‘next’ button, the search will be initiated.

2.9 CSearchPage

This class is derived from the standard MFC class CPropertyPage. This page displays the product ids as returned by the mermaid search. In order to add a product to the basket, the user checks the check box next to the name of the product. Double clicking on an entry, or selecting a product and then clicking the ‘View Details’ button displays the View Products Dialog CDlgViewProduct. Pressing the ‘Next’ button adds the selected products to the basket at the mermaid server and proceeds to the billing details page.

2.10 CBillingDetailsPage

This class is derived from the standard MFC class CPropertyPage. This page provides the user with fields to enter the user’s name, organisation name, Billing address city, street, country, post code and phone number. No field may be left blank. Pressing the ‘Next’ button sets the CMermaidBillingAddress m_billingAddress member variable in the CBMTDatabrokerApp class. Pressing the ‘Next’ button progresses to the credit card details page.

2.11 CCreditCardPage

This class is derived from the standard MFC class CPropertyPage. The page allows entry of the user’s credit card details – cardholder first name, last name, card number, card type, expiry month and expiry year. None of the fields may be left blank. Pressing the ‘Next’ button sets the CMermaidPayment m_payment member variable in the CBMTDatabrokerApp class with the corresponding payment details. Pressing the next button progresses to the next confirm delivery page.

2.12 CConfirmDeliveryPage

This class is derived from the standard MFC class CPropertyPage. The page allows entry of the online delivery address for the purchased product. The fields prompted for entry are FTP address, FTP User name and FTP password. Pressing the ‘Next’ button progresses to the Order confirmation page.



2.13 CConfirmOrder

This class is derived from the standard MFC class CPropertyPage. The page displays a short summary of the order. To confirm the order the 'Next' button is pressed. This completes the order and initiates the download. This takes the user to the next page which displays the download status of the initiated downloads.

2.14 CDownloadStatusPage

This class is derived from the standard MFC class CPropertyPage. It displays a list of products displayed against the percentage complete status of the downloads. If the user opts to exit the application the downloads will continue. If the application is left activated then once all downloads are complete, the signalled state of the IBMTNotify communication object will be set.

2.15 CDlgViewProduct

This class is derived from CDialog. It has member variables storing the values corresponding to the dataset details retrievable by the mermaid interface component. The information displayed to the user are Product Name, Product ID, Product Provider, Product Price and Product Description. This is simple non-interactive popup dialog.

2.16 CMermaidInterface

This class is used as a wrapper class to the mermaid interface component. It is used to allow function calls using the wrapper classes defined for use in the BMTDatabroker application.

2.16.1 Methods

2.17 CMermaidBillingAddress

This class wraps the _BillingAddress interface exposed from the mermaid interface component dll.

2.18 CMermaidPayment

This class wraps the _Payment interface as exposed from the mermaid interface component dll.

2.19 CMermaidSearchParams

This class provides a container for the search parameters that will be passed to the search product method of the CMermaidInterface class.