Chapter- 2

Components of Data Providers

प्रत्येक .NET डाटा प्रोवाइडर (Data Provider) System.Data नामक नेमस्पेस में समाहित होते हैं, तथा प्रत्येक के अपने अपने कुछ classes होते हैं| प्रत्येक। NET डाटा प्रोवाइडर के निम्नलिखित चार प्रमुख कंपोनेंट्स होते हैं

- Connection : इसका प्रयोग करके हम अपने वेब एप्लीकेशन को विभिन्न डाटा सोर्स से जोड़ने के लिए किया जाता है|
- Command : एक बार जब डाटा सोर्स से कनेक्शन बन जाता है तब कमांड ऑब्जेक्ट के जरिये हम डाटा सोर्स पर विभिन्न operation execute कर सकते हैं जैसे Insert, Update, Delete|
- DataReader: इस ऑब्जेक्ट का प्रयोग हम डाटा सोर्स के डाटा को वेब एप्लीकेशन पर दिखाने के लिए करते हैं| यह एक readonly resultset होता है अर्थात इसके जरिये सिर्फ डाटा को read कर सकते हैं उसमे कोई सुधार नहीं कर सकते हैं|
- DataAdapter : इसका उपयोग डाटा सोर्स के डाटा से DataSet को populate या भरने के लिए किया जाता है|

Connection Object

यह ऑब्जेक्ट डेटाबेस के लिए connection का निर्माण करता है| .Net फ्रेमवर्क के अंतर्गत कई सारी connection class उपलब्ध है जैसे SQLConnection और OleDBConnection Object| SQLConnection Object का प्रयोग माइक्रोसॉफ्ट के SQL Server के डेटाबेस से जोड़ने के लिए किया जाता है जबकि OleDBConnection का प्रयोग MS Access, Oracle जैसे RDBMS से जुड़ने के लिए किया जाता है| Connection Object बनाने के लिए निम्नलिखित जानकारी मुहैया कराई जाती है

- DataSource Name
- File name
- username
- password
- Port

Command Object

जैसा की आम समझ ही गए होंगे की कमांड ऑब्जेक्ट का प्रयोग डेटाबेस पर विभिन्न SQL कमांड execute कराने के लिए किया जाता है जैसे अगर आप डेटाबेस की किसी टेबल पर रिकॉर्ड जोड़ने चाहते हैं तो कमांड ऑब्जेक्ट का प्रयोग करके हम insert SQL क्वेरी का प्रयोग कर लेंगे| जैसे कई प्रकार के डाटासोर्स से जुड़ने के लिए कई कनेक्शन ऑब्जेक्ट होते हैं उसी प्रकार कई प्रकार के कमांड ऑब्जेक्ट भी .NET फ्रेमवर्क में उपलब्ध होते हैं| जैसे SQLCommand object जो कि SQLConnection की साथ काम करता है
वहीँ OleDBCommand object OleDBConnection के साथ कार्य करता है|

DataReader Object

डाटा रीडर ऑब्जेक्ट यह ऑब्जेक्ट डाटाबेस से एक फॉरवर्ड ऑनली, रीड ऑनली, कनेक्टेड स्ट्रीम सेट प्रदान करता है। डाटा प्रोवाइडर की अन्य घटकों के विपरीत, डाटा रीडर ऑब्जेक्ट नहीं बनाई जा सकते हैं। सच तो यह

है कि डाटा रीडर कमांड ऑब्जेक्ट की ExecuteReader मैथड के परिणाम के रूप में लौटाया जा सकता है। SQL Command Execute Reader मैथड एक SQL डाटा रीडर लौटाता है तथा OleDbCommandExecute Reader मैथड एक OleDb डाटारीडर ऑब्जेक्ट लौटाता है|

जब आप मेमोरी में डाटा को छिपाकर नहीं रखना चाहते हों, तो डाटा रीडर एप्लीकेशन के लॉजिक के लुए सीधे ही डाटा की पंक्तियाँ जुटा सकता है| क्योंकि मेमोरी में एक समय में केवल एक ही पंक्ति होती है, अतः डाटारीडर सिस्टम परफार्मेंस के विचार से निम्नतम ओवरहेड की व्यवस्था करता है, किन्तु अपने लाइफटाइम के लिए किसी खुले कनेक्शन ऑब्जेक्ट का विशिष्ट (excusive) उपयोग करता है |

Data Adapter Object

डाटा एडॉप्टर ADO.Net के डिस्कनेक्ट डाटा एक्सेस के बीचो-बीच स्थित क्लास है | यह वस्तुतः डेटाबेस तथा डाटासेट के मध्य सभी संपर्क की सुविधा प्रदान करने वाला मध्यस्थ है | डेटा एडॉप्टर का प्रयोग या तो किसी डाटा-तालिका को भरने के लिए, या फिर डेटाबेस से इसकी fill मैथड द्वारा डाटा लेकर डाटासेट को भरने के लिए किया जाता है | मेमोरी में रहने वाले डाटा को मेनिपुलेट कर लेने के बाद, डाटा एडॉप्टर 'अपडेट' मैथड को कॉल करके डेटाबेस में परिवर्तन कर सकता है | डेटा एडॉप्टर निम्नलिखित चार प्रॉपर्टीज प्रयोग करता है जो डेटाबेस कमांड्स का प्रतिनिधित्व करते हैं :

- Select Command
- Insert Command
- Delete Command
- Update Command

जब 'अपडेट' मैथड कॉल किया जाता है, तो डाटासेट में किया गए परिवर्तनों को वापस डाटासेट के लिए कॉपी कर लिया जाता है, तथा उपयुक्त 'इन्सर्ट कमांड, 'डिलीट' कमांड या 'अपडेट' कमांड एक्सीक्यूट की जाती है |

Difference Between Datasets & DataReaders

ADO.NET Framework के क्लासेस (classes) दो भिन्न डाटा एक्सेस मॉडल्स (data access models) का समर्थन करते है| आप **DataSet** या **DataReader** का प्रयोग कर डाटा को प्रतिनिधित्व (represent) करने के लिए ADO.NET के क्लासेस (classes) का प्रयोग कर सकते है |

एक DataSet, मेमोरी (memory) में स्टोर्ड (stored) डेटाबेस को दर्शाता है ; क्योंकि जब आप किसी डेटाबेस के डाटा को एक DataSet से वर्णित करते हैं,तो आप फिजिकल डाटाबेस (physical database) को मेमोरी (memory) में वर्णित किया जाता है |

दूसरी तरफ DataReader एक समय में किसी डाटाबेस के केवल एक ही रिकॉर्ड (record) को दर्शाता है | अतः जब आप DataReader का प्रयोग करते हैं तो आपको अपने एप्लीकेशन में डाटाबेस के सभी रिकार्ड्स (records) को रिट्रीव (retrieve) करने के लिए एक लूप (loop) के माध्यम से एक-एक कर रिकॉर्ड (record) को एक्सेस (access) करने की आवशयकता होती है |

DataSet का DataReader के तुलना में प्रयोग करने के दो लाभ हैं | पहला लाभ यह है कि Visual Studio.Net, DataSets का बेहतर समर्थन करता है और यह DataSets को जेनेरेट (generate) करने और इनके साथ कार्य करने के लिए कुछ टूल्स (tools) मुहैया करता है | दूसरा लाभ यह है कि DataSets को मेमोरी (memory) में कैश (cache) किया जा सकता है | अतः आप एक DataSet को सर्वर (server) की मेमोरी (memory) में स्टोर (store) कर सकते है तथा उस DataSet को विभिन्न पेजों (pages) से आए हुए रिक्वेस्ट्स (requests) को सर्व (serve) करने के लिए रियूज (reuse) कर सकते है | मेमोरी (memory) में डाटा को कैशिंग (caching) करने से आपके एप्लीकेशन (application) का परफॉरमेंस (performance) बढ़ जाता है |

DataSets के प्रयोग की कुछ हानियाँ (disadvantage) भी हैं | पहला यह कि,जब आप किसी डाटाबेस के रिकार्डों (records) को एक DataSet से रिप्रेजेंट (represent) करते है तो सभी रिकार्डों (records) को वेब सर्वर (Web Server) की मेमोरी (memory) में कॉपी (copy) करने की आवश्यकता होती है | दूसरा यह कि DataSet, DataReader की तुलना में स्लो (slow) होते है | अतः जब आपको किसी डाटाबेस-टेबल (database-table) से रिकार्डों (records) को किसी पेज (page) पर प्रदर्शित करना हो तो DataSet के बजाए DataReader का प्रयोग श्रेयष्कर है |